

Healthcare In the Digital Age

The future of health records

A thesis by Eliot Slevin

Dear Reader.

This thesis was written and designed, to be consumed as a website. Please visit <http://barnett.surge.sh/>

I highly recommend reading the web version, as it includes features and nuance which were not designed to be re-created in print. This print copy exists as an *archival copy*, while I have done my best to ensure it's as close to the website as possible - in places the structure, wording, and content are just not as good.

There are two ways to view the website - the simplest is to visit <http://barnett.surge.sh/>

If that URL is unaccessible, please request the digital files submitted alongside this thesis. They contain a full copy of the website, and instructions on how to run the website from your own computer.

A 90-point thesis submitted to the Victoria University of Wellington in partial fulfilment of the requirements for the degree of Master of Design Innovation in Media Design.

Victoria University of Wellington, 2017
School of Design

Acknowledgements

This thesis wouldn't be the same without the people around me. I'd like to thank:

My supervisor Walter Langelaar for never saying an idea is too big.

My girlfriend Lucy McMaster for her support and motivation.

My mother Cathryn Shine for her love.

My father Grant Slevin for his wisdom.

And my sister Serafina Slevin for her brownies.

Table of Contents

1. The Paradox of Healthcare Software	5
2. What actually is an EHR?	8
3. Interviews	12
3.1 Olivia	14
3.2 Paul	16
3.3 Michael	18
3.4 John	21
3.5 Will	26
3.6 Bryan	28
4. Survey	32
5. Complexity & Variety	35
6. Risk	37
7. Distribution	39
8. Brief & Design Methodology	41
9. Storage	43
10. Network	47
11. Authentication and Privacy	53
12. User Interface	58
13. Output	63
13. Critical Review	65
14. Bibliography	67
15. Appendix	71

Abstract

Electronic Health Record software (EHR), is used by medical professionals regularly to interact with patient records. The functionality of this software is key to public health, however the quality of this software does not match it's importance, or its cost.

Developing innovative healthcare software is a difficult due to the inherent challenges of complexity, risk, and distribution faced by healthcare software.

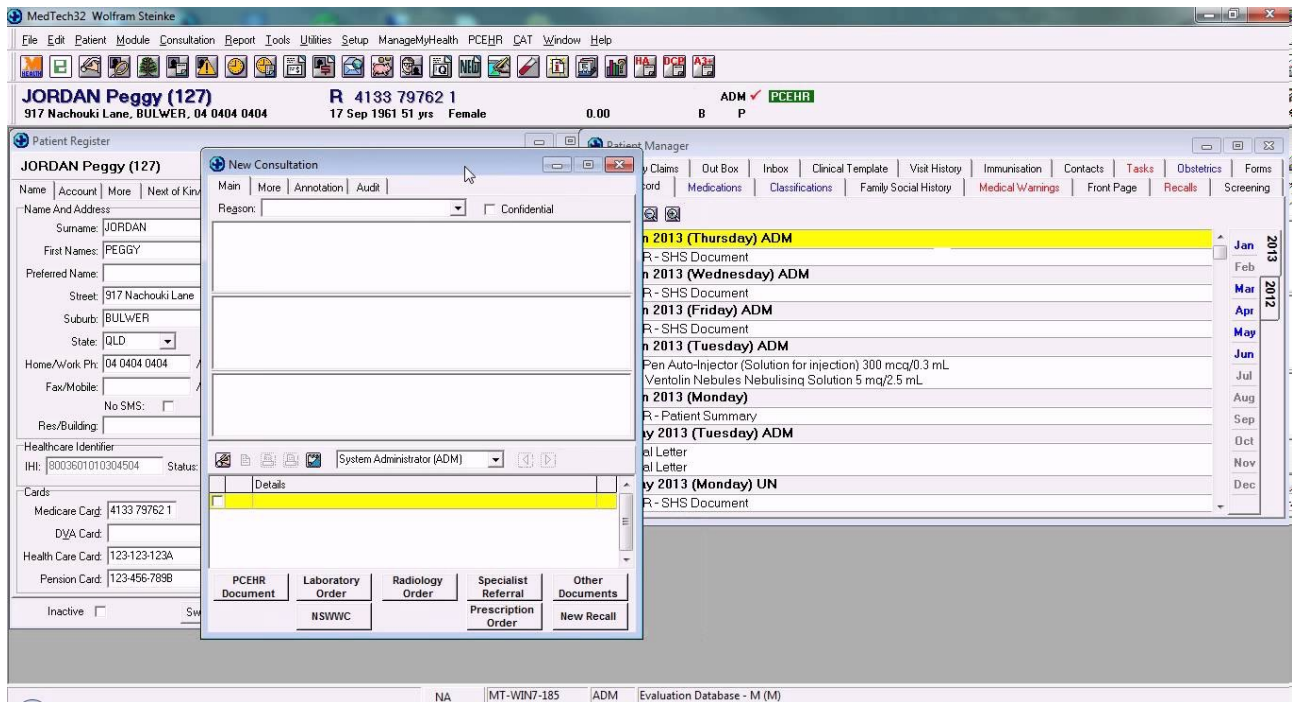
In response these challenges, this thesis proposes Barnett: a novel system to store, share, and interact with health records across institutions. Notably, this system moves control from the vendor to the user - through iterative, crowd based improvement, and ownership. This allows the system to fit unique, and varied user needs.

Barnett was developed through an interdisciplinary qualitative research process, grounded in perspectives from design, software engineering, healthcare; and interviews with healthcare professionals.

The survey of current EHRs, healthcare models, and the design process indicates that developing systems which enable a faster iterative cycle of design, development, and distribution is potentially a more sustainable approach to electronic health records.

The Visit

I've always found going to the doctors a strange experience. Usually, it's a sore throat, sometimes something worse. I sit down, start explaining my problem to my doctor. They listen patiently and start taking notes, with this piece of software.



MedTech 32, Practice Management Software

This part always bothered me - because the software they use doesn't seem that great.

I'm a software developer and interface designer, so I am more sensitive than most. However, from my perspective as a patient, it never filled me with confidence. The software looks like it hasn't been upgraded since Windows XP. It doesn't look that easy to use. It makes me question how safe my data is. And most importantly, my medical record doesn't feel like mine. It's owned by that piece of software.

The bizarre part is, the doctor is fantastic. They have trained for over half a decade, have a wealth of knowledge and experience, and are supported by the most amazing technology. Decades of drug research and advancement, only a prescription away. Almost any biological sample you can produce, blood, urine, swabs, can be tested for disease. X-Rays, MRI scans, Ultrasound - they can gain a perspective of you nobody else can. They are such a contrast to the tool they use every single day, for every single appointment - which appears to have completely missed this technological advancement.

It turns out I'm not the only person to think this - every doctor I have talked to has been dissatisfied with the state of healthcare software. Some doctors are even writing songs about how bad their software is (ZDoggMD, 2015).

"They built you a turd, doc" - ZDoggMD

The Challenge

It would be foolish to think nobody has tried to solve this issue before. The truth is, many people have tried to improve healthcare software, and many have failed.

In 2002 the UK Government decided they needed to build an Electronic Health Record (EHR) system for their citizens. They dreamed of connecting all of their healthcare institutions together, from primary care (community doctor's clinics) to hospitals, to pharmacies. This system was going to store patient records, manage electronic prescriptions, handle patient appointments and more. It was going to revolutionise the UK's healthcare system.

It was a complete failure. The project was originally going to cost £2.3 billion (\$7.8 billion NZD), and take three years. However, nearly ten years later, and £12.4 billion (\$27.2 billion NZD), their goals were not met (National Audit Office, 2006). They had failed to finish most of the project, and the parts they had weren't getting any real world use. The majority of their investment was written off, wasted, and the project was shutdown in 2011 (Department of Health, 2011). It was arguably, one of the largest IT failures in the world.

It can be hard to comprehend how much a billion is. But considering that a qualified nurse starts at around £21k per year (\$36k NZD), over the ten years they could've spent that money on 44 thousand extra nurses.

Of course, perhaps its easy to see how a bureaucratic government process could chew through money. Maybe they needed to leave it up to a competitive market, and let the best software win? This was exactly what America did (Robert, 2015). They decided to provide a stimulus package of \$30 billion USD - to pay software vendors to improve their software, and pay institutions to start using it.

Their goal was to computerise the healthcare system, and the usage of computers did increase dramatically. However, the quality of their software, Robert argues, is not worth \$30 billion USD. They now have a system where there are many different products, all with different formats, and getting them to cooperate is a huge challenge. The previous director of America's health IT office (before the stimulus package), said:

"We've built the Frankenstein I was most afraid of" - David Brailer

Private institutions in America have also faced huge costs when implementing large Health IT projects. Hospital chain Kaiser Permanente, spent 10 years and \$4 billion dollars on their EHR system - about \$444 per registered patient (Snyder, 2013).

A smaller chain Partners spent \$1.2 Billion on their EHR - a cost of about 2 million dollars per doctor (McCluskey 2015). While both these institutions found success with their new software, the cost is still undeniably, immense.

The Question

The people who have tried and failed had a lot in their favour. They had experience in the healthcare industry. They had institutional support, from both medical institutions and government institutions. They had political power, and influence. And they had lots of funding, which they could use to get the absolute best people they could find.

But, with all that, they all had issues with cost, quality, or complete failure.

It makes you wonder - what were they missing? Why did they fail? Why did it cost so much? And the larger question - **Why is healthcare software so hard to build?**

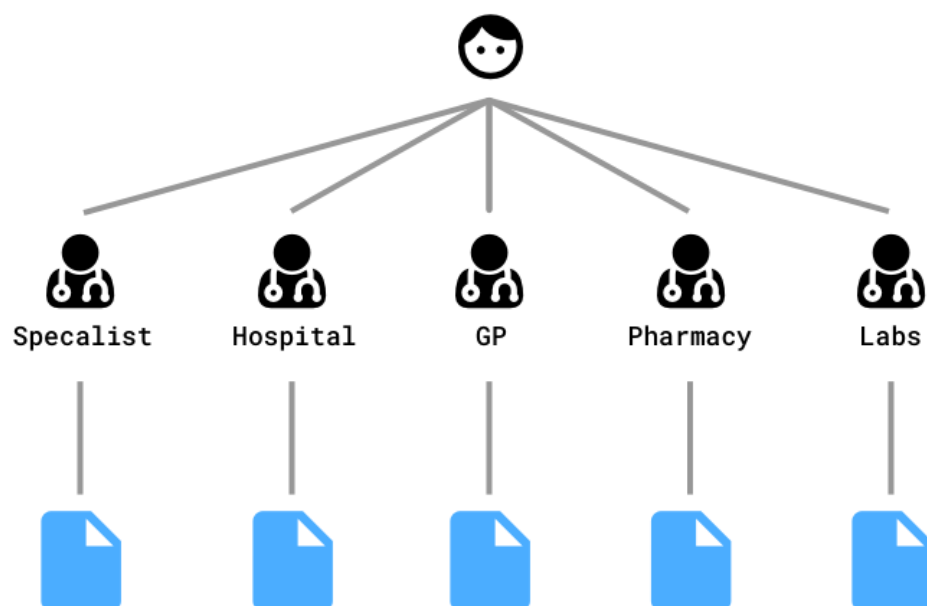
That is the topic of this thesis. I've spent the last 12 months trying to answer that question, while designing a radically different Electronic Health Record (EHR) system for primary care in New Zealand. But first, let me explain what an EHR system actually is.

What is an Electronic Health Record

In New Zealand, your medical records are accessed like this. You go to your doctor, tell them your sick. They open a piece of software and add some information to your record.

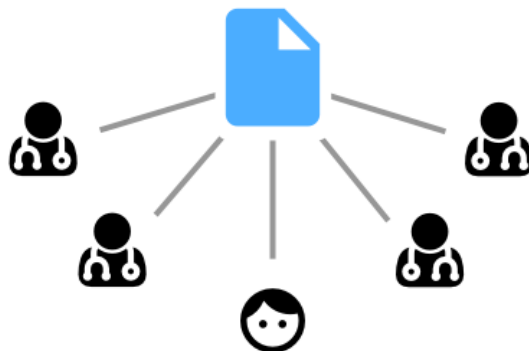


However, almost all patients go to more than one healthcare institution. An average patient could visit a GP, the hospital, and a specialist. On top of that, there are records about you kept at institutions you might not think of - your pharmacy, and the labs which your GP refers you to. So the model actually looks more like this.



These records are considered *medical records*, and the software which manages them is an Electronic Medical Record System. These are not Health Records. The distinction is that these records only show a slice of your health - your GP doesn't know what your specialist knows, and they don't know what the hospital knows. All of the institutions only are seeing what information is relevant to their institution, not your actual *health*.

An Electronic Health Record, however, works like this:



All interested parties have access to a centralised record. This record takes the needs of all institutions into consideration, and you. This means it can be considered a health record, as it aims to assess your health from as many facets as possible (Amatayakul, 2009).

An EHR system is more than a storage system, it's a network. If you visit a hospital, they already have a list of medications you've been prescribed on hand. If you have to visit a different GP (say you're on holiday), they can just look up your record in the EHR system. Your records are accessible when and where you need them.

Practice Management Software

The software used in General Practices in New Zealand are categorised as practice management software (PMS). They deal with the day to day operation of the medical practice.

This includes an EMR system, but also appointment scheduling, invoicing, managing government claims, patient analysis, and more.

New Zealand has a few PMS products in usage, with MedTech32 holding the market majority:

- MedTech32
- MedTech Evolution
- MyPractice
- Profile (For Mac and Windows)
- Indici (Web App)

Scope of research

The question I want to explore is, "What are the difficulties faced in developing healthcare software". However, the scope of this research is limited, due to time. So, I'm going to focus on a few specific areas:

EHR Systems. There are a lot of different categories of healthcare software, but I will be focusing my research on EHR systems.

Primary Care. New Zealand citizens mostly interact with Primary Care - General Practices, and Pharmacies. The alternative is Secondary Care - Hospitals, and Specialists. The majority of my focus is going to be on Primary Care, as that's where the majority of healthcare takes place.

New Zealand. It's much easier to design in a system I'm familiar with.

Finding research which fits all of these criteria is, well tricky. So this is not a strict scope. However, in terms of my design output - it will be focused on these criteria.

Research Methods

To research this question I'm focusing on three methods - A cross-disciplinary literature review, qualitative, semi-structured interviews, and a survey (Martin & Hanington, 2012).

These research methods are most appropriate due to the scope of this project, and the scope of my output. Creating a practical EHR system would require years of time, lots of funding, and a team of programmers, designers, and medical professionals. Even if I could create a system, without backing and buy-in from institutions, it would get minimal real world usage.

So, what is the value of my thesis? The best use of my time is to: discover problems, and generate ideas. Unlike commercial vendors, I don't have the requirement to make a profit - so I can explore ideas and problems which may have been dismissed due to being financially unfeasible.

Therefore, I picked my research methods with the aim of finding the unexpected and trying unique approaches.

Cross-Disciplinary Literature Review

Healthcare software exists at the intersection of many fields - software development, design, and healthcare. Beyond that, it bleeds into the fields of media theory, business, governance, and economics. So it only makes sense to approach my literature review from every discipline. There are many different ways to approach the problem of software for healthcare, so I wanted to review all approaches.

In a traditional thesis, at this point I'd summarise themes from my literature. However, as my medium is a website, I can approach this differently. My literature is spread throughout my entire design section, and I discuss specific works when appropriate.

All the literature reviewed is linked together on this index page, where they're organised by field.

Interviews & Survey

My primary user research method was semi-structured, face to face interviews. I used a semi-structured approach as I was specifically looking for topics I wouldn't have thought to ask about.

As additional user research, I performed an open-ended survey with 19 NZ GPs, which was a very effective way to discover major themes and complaints.

Both methods are discussed in further detail in the next section.

Interviews

My main user research method was semi-structured interviews (Martin & Hanington, 2012).

Wherever possible, I interviewed face to face. However, due to location, some interviews were via Skype or phone call - whatever was convenient for them.

All of my interviews have been confidential - all of their names have been changed, and any information that could identify them has been removed. This was to ensure they could be completely honest, without concern of our discussion impacting their lives.

The interviews were semi-structured, I had a list of questions which were designed just to be talking points - my main interest was to find out as much about them as I could, particularly things I would've never thought to consider. For each person, I did background research and tailored the talking points to fit.

I was willing to interview anybody who had experience or knowledge about medical software in NZ. This ended up being a fascinating interesting mix of people, from daily users such as GP's and Nurses, founders of healthcare software companies, to the highest level of healthcare management in the NZ Defence Force. My goal with these interviews was to find topics and points I wouldn't have found otherwise, and I certainly succeeded.

I had the pleasure of interviewing some very interesting people, and I would like to thank them again for their time.

Interview Overviews + Key Takeaways, full summaries following.

Olivia - Managing nurse at large community clinic

77% of their patients use their online patient portal, and the first patient to use it was 80 years old. The majority of annoyances with her software come from the interface - otherwise pleased with the functionality and simplicity of MedTech32.

Paul - Possibly the remote GP in New Zealand

He is the only doctor for a huge region in the south island - and he also runs the ambulance service. He is a beta tester and active participant in his software's development - in email contact with the developers weekly.

Michael - Founder of healthcare software company, with a background in psychology.

In his opinion, the biggest risks faced in healthcare software is human action and financial failure. He has found a lack of institutional support for innovative, good healthcare software in New Zealand.

John - John is in charge of the entire New Zealand Defense Force's healthcare.

They reduce running costs dramatically by avoiding a doctor-centric model. The Defence Force is investing in the future - through large-scale analytics, and 24/7 monitoring. They have both the resources, and culture to implement change.

Will - AI powered EHR systems

AI to structure healthcare information is in serious development. This technology has the potential to revolutionise Health IT. EHR systems are hard to build due to disagreements among GPs.

Bryan - The process-obsessed GP

Quality processes are necessary to keep on top of GP's large workloads.

GP's can be incredible power users. Dissatisfied with MedTech - but willing to find workarounds.

Good, consistent documentation is key to good practice.

Olivia

Nurse manager at a large community medical centre. This means beyond being a nurse she helps organise and manage the other nurses. She also helps ensure that their practice is meeting certain health targets from the Ministry of Health, eg cardiovascular, smoking, cervical smears, immunisations.

The Practice

The practice is a large community medical centre, in an area of mostly families and children. They interact with patients three ways:

- Face to face, traditional appointments
- Over the phone
- Through an online patient portal

"70% of our adult population is signed up with the patient portal ... The first patient we signed up the portal was 80, he's still using it"

I was very surprised how popular the patient portal was. They use it's functionality extensively; patients can view their record, view their lab results, make appointments, request prescriptions, and even message their GP.

At first the patient portal was 'a real pain', but the GPs are getting better at telling patients that they'll have to make an appointment rather than just emailing the doctors. It's working well for them now and is popular among the community. They are currently not charging for the portal, but they're currently assessing how they will price it.

The Internet at the practice isn't the best - it's annoyingly slow, and goes down quite often. They use a digital record transfer service GP2GP to transfer records, and they receive their lab results electronically.

Software

They use MedTech 32, but are currently looking at upgrading to MedTech Evolution. MedTech is used for more than just health records, it also is used for administration tasks (e.g., billing, invoicing, booking appointments, patient population statistics, etc.). It's used by both doctors and the Admin staff.

As a nurse manager, Olivia spends about half of her time in MedTech dealing with patient records, and half dealing with administrative tasks.

Annoyances

The chief complaint is that MedTech is slow to update/fix things. She gives the example of immunisation schedules. Whenever there's an update on an immunisation schedule (which happens fairly often), they don't remove the old schedules. So when a high-risk child comes in, and they need to be on a different schedule, the list is so huge and there are so many choices that it's hard to know which ones are current, and it's easy to make a mistake. Some schedules they just don't use, but they can't remove them.

She even contacted MedTech to remove some of the old schedules, but nothing has been done.

In another part of the software, there is a huge list she has to select from. However, it doesn't provide any way to filter/jump to parts in the list - this is incredibly annoying as she has to scroll through all of the options to find what she's looking for.

They also use a few web-based integrations into MedTech - one is for clinical decisions support, another is to check lab results online. However, as their Internet isn't very fast, it's frustratingly slow. A few of the GPs even refuse to use them, as they're so slow.

Good things

One thing they really like about MedTech is that it's simple and easy to use. They have some new nurses starting in a few weeks - they haven't used MedTech before, but Olivia is confident they'll be able to pick it up.

Despite their integrations are annoyingly slow, they like that they have the option to integrate other systems into MedTech. The flexibility is good for their practice; however, there are a few systems that they want, which can't be integrated into MedTech.

Feedback

Olivia was not concerned with the idea of multiple healthcare staff collaborating on one record, provided it was secure. She viewed anything which increased accessibility as good for patients.

She thinks if a PMS could be designed which can easily integrate with other software, that would be good.

Paul

Paul is potentially the most remote GP in New Zealand, and a beta tester for Profile for Mac.

The Practice

He runs a sole practice in a very rural spot in the south island. The patient population is pretty normal, but in terms of geographical area served, he thinks it might be the largest. On top of being the only GP for kilometres, he also runs the ambulance. He runs 15-minute appointments, and the day I spoke to him he had seen 26 patients. The Internet quality is poor, so he does not depend on any Internet services.

Profile for Mac

He has experience with MedTech32, but he uses Profile for Mac - and loves it. Profile doesn't have the market share among NZ GPs, but Paul thinks it's easily superior. Profile provides a very similar feature set to MedTech. He uses it mostly for appointment management, patient records, task management, and invoicing.

He is such an incredible fan of Profile for one reason - the user interface. He found MedTech's user interface, and user experience "clunky", annoying, and a waste of time.

For example, one UX feature of profile he loved was the tagging system. Visitation notes can be tagged against certain patient problems. So when you view a patient with say, cardiovascular risk, you can click the problem, and it will show all visitation notes related to that problem. Profile also has impressive graphing features, and patient population analysis features.

He handles his own backups to his own hardware, and can restore a backup in 20 mins.

His only complaints about his software are that there is only one developer working on the Mac version, so updates are slow, and minor UI tweaks he'd like. I asked him if he had a magic wand, what would he change about his software, he said:

"I'd give every GP in New Zealand Profile for Mac."

Beta Testing

The most interesting thing about Paul is that he is one of a couple of doctors that beta test Profile. He is sent pre-release builds of the software, and he helps them find bugs. They have an incredibly tight feedback loop - they'll send him a new version in the morning, he'll find six bugs by lunch time. They appear to be in contact every three days or so. Being a sole practice, it's much easier for him to test and try new tools.

He takes incredible pride in his tools, which includes his software, which he thought was unique among most GPs. Beyond helping test and influence the future of his software, he helps debug for

other GP's he knows who are using profile. They'll send him emails, and he'll diagnose the issue, and send instructions to fix it in just a few hours.

Collaboration

I asked him his opinion on HealthOne, the shared care system which runs in the south island. He had his concerns that patients hadn't been properly notified of how their information was shared, and thought that it shouldn't be presumed that patients will want their data shared. However, he did say the utility of the system was great - getting the details was really good.

He noted that it is sometimes hard to make sense of another doctor/nurses notes.

Michael

Micheal is the founder of a healthcare software product, and has a background in clinical psychology & research.

The Product

Micheal's product helps manage the mental health of at-risk individuals - people suffering from depression, alcohol abuse, etc. The main focus of the product is case management - which fills the same role as medical record software, but for mental health. Case files include very similar things to health records - session notes (visitation notes), communications, and action plans. His product also provides a screening service - which lets you quickly check patients for any risks (e.g., depression, anxiety, drug dependencies, alcohol dependencies).

The product is used both in healthcare, but also in welfare. It's used in both clinical settings, hospitals, and by the NZ Police. It's used across sectors as mental health can easily be affected by welfare - e.g., dysfunctional families or poverty can affect depression, anxiety, or suicide risk. The New Zealand Police use it to screen people for Alcohol Use Disorder.

Why build this product?

He started by writing best practice guidelines and books for clinical psychology, but the implementation and uptake of the practices were patchy. So they decided to focus on implementing best practice into the software itself. This helps ensure that it happens, and is done correctly - and by individuals without nuanced training, particularly at the initial stages.

They decided to build the tool as when they started over a decade ago; there wasn't anything on the market which fitted their needs. There was software for use in a clinical setting (such as a hospital), but it was rudimentary, and faster to just do on paper. But most importantly they needed something which could be used by social workers, in the context of welfare.

They saw the opportunity to build an innovative product which ensured best practice would be followed, and blend the world of welfare and mental health.

Biggest challenges in building this product?

The first challenge was ensuring you have the right people on your team - people who will take the risk, and stay with you on what is a long, difficult road. Financially, innovative healthcare software isn't super profitable, and they have run out of money a few times.

The second challenge was a lack of institutional support. Both regarding financial support, and other things such as business development, marketing etc. He said it's common to find people who want to use your product, but its hard to find people who want to support the continual development of it.

For example, they started as a research project at a New Zealand University, but as they were dealing with active patients, it was seen as better if it was a separate company. This meant they couldn't benefit from the University's innovation support programs. The New Zealand Health IT Board was nice to him, but provided no help - the board is dominated by individuals who are part of large health IT companies.

He does not see it as a level playing field, and there is nothing to help facilitate and support the development of innovative ideas in healthcare. They have received much more institutional support from the welfare sector, such as ACC & Police Departments.

Multi-Institutional

One innovative feature of the product is that it's designed to be used across institutions - the case file is shared with all individuals working on the patient's case. They all have the same view, and nobody is in the dark about information. I asked him why he specifically designed it like this.

He had two reasons - firstly, he has worked across both welfare and healthcare sectors, and was tired of the 'ping-pong' treatment of patients - being referred from institution to institution. Some people justify it with an order of treatment - e.g. you can't treat depression until you treat the alcohol disorder, but it's just untrue, and the literature now agrees.

Secondly, when they started the idea of a client-centric management was only becoming popular. It has yet to become a key theory, but they think it will. They have also been looking at research which shows that the patient themselves should be an active participant in their case, not just welfare/health workers.

Their product is designed at its core to work with multiple users. One feature they're developing is a holistic graph of a patient's needs - so you can visually see the state of their health, housing, mental health, family environment, etc. This is designed to give everybody in contact with the patient perspective of where the gaps are.

Challenges of Multi-Institutional

He noted that variety was a challenge of designing it this way - different institutions have different ways of doing things, and different ways to record information (and different standard information requirements). He noted variety is huge in healthcare, and it's probably only going to get worse as more technology advances. They simply have to work with institutions to find solutions. They have developed a principle for development - if a user needs some features built, they'll make it available for all users. This changes the relationship from a customer, to partner - and promotes their users to work together to finding funding for common needs.

The other challenge was patch protection - often big vendors try to hold onto a specific type of institution (eg MedTech in primary care). If you're building a product to work across different types of institutions, you have to compete with a larger variety of big companies. He noted that the products with the largest market shares were fantastic when they first came out - but they don't keep up and get old, while pretending to stay innovative. He described this quite succinctly:

"The gentrification of software"

He did note that the people purchasing software kept using old software because they can be confident in its functionality. They are acutely aware of the risk of new software, or systems - but they have traded confidence for a lack of innovation.

Managing Risk

He had two points to make about managing risk:

When they were first starting out, the developer they were working with locked them out of the running system and demanded more payment. They were totally caught, as they were managing a population of suicidal kids. Later on, they followed the legal process and ensured action was taken, but they initially had to pay the money to get access back, as there was nothing they could do. This taught them the lesson that you need to work with really ethical people, and they need to protect how the system is deployed and running - including using multiple developers.

His other point was that at the end of the day nobody is going to help you (financially). You need to be prepared to carry on for nothing, or know to give up.

I found these answers most interesting out of the entire interview. Managing risk in the world of software development means reducing the chance of failure, or error - but in Micheal's own experience the biggest risk has nothing to do with that - they were human action, and financial failure.

Insight

Their product does a fantastic job at turning information into insight which is relevant to the user. I asked him how do you know how to do this, he said it comes down to experience and research.

John

John is in charge of the entire New Zealand Defence Force's healthcare. He has a background in emergency medicine and general practice. In a prior life, he was an engineer.

The New Zealand Defence Force is made up of the army, navy, and air force. Together they have a total of 14.2k personnel.

The main reason the Defence force has a healthcare system is to keep their personnel healthy, and fit to do their job - without putting strain on the public health system. They also provide healthcare in aid situations - they will go to international disasters and provide help and care.

John's Role

John has two main jobs within the Defence Force:

1 - Make people fit to serve - this is slightly different to what is provided by a GP in public healthcare. If you have an issue which isn't solved but doesn't affect your ability to serve, it's not a problem for them.

They are a healthcare provider in a business whose core business - is not healthcare. They are an enabler for the main function of the Defence Force - so he has to work around them.

They run essentially all of what is provided by a DHB. They provide a full primary care system for their personnel. Additionally, they have the complication of having to move it - wherever their troops go; they need to provide healthcare.

"Imagine picking up Auckland DHB, moving it to Dunedin, then three weeks later moving it to Fiji."

2 - Collect information. He needs to be able to show leadership at any time how many troops are fit to be deployed - and what the long term societal burden of service is. When a person is deployed, has exposure, that creates liability. They need to look at the longitudinal information to inform both the Defence Force and Government.

Defence Force healthcare model

John describes healthcare models as coming in a couple of ways - usually sick-care vs wellness model. Sick-care is what most people are used to - you get sick, you go to your GP, they have a solution. Wellness models focus on keeping healthy people healthy - for example, immunisations, proper diet, contraception, public health education are all examples of well-care.

The Defence Force aims to sit half way. They provide sick-care, but they also provide well-care - as they need to ensure everybody is still fit to serve.

For example - if you're part of a healthy civilian population, providing you don't have a chronic condition, you would see your doctor 2-3 times a year. However, if you're part of the Defence Force you would make 11 points of contact a year.

John's healthcare team provides 110k patients contacts a year, usually for a simple check-up. This allows them to catch problems when they're small, and provides a huge overview of the population - every contact collects data.

Implementation

The most amazing part of their system is that they provide these 110 thousand patient contacts per year, with a team of only 24 registered doctors. They do this by not using a 'doctor centric' model.

Their doctors provide the ability to manage complex cases, and they guide and govern about 850 additional healthcare staff - who do about 2/3rds of the workload. This includes nurses, and medics - personnel trained in simple healthcare, but not registered.

In civilian healthcare, the nurses operate under a limited scope of practice. However, in the Defence Force nurses and medics can examine, diagnose, and treat - all without a doctor being involved.

The Software

They use Profile for Windows. MedTech doesn't work for them - it's good for a linear model of care, and small practices - but that's not how they work.

He described the UI of Profile as clunky, but the database and backend systems as 'phenomenal' - it was one of the main reasons they chose Profile. The volume which they're using Profile (and will be in the future), has made it a worthwhile choice.

The size they could potentially have to scale to is immense. They could go from all of the Defence Force - to all of New Zealand - to all of The Coalition if needed. Seems extreme, but it's a situation they have to consider.

I asked him if they ever considered making their own software, instead of using a vendor. They have connections with many other militaries around the world - and there has been a number who have tried to self-make, and he has yet to see a successful case. There have been examples where countries put vast money into systems - in one case, more than an entire Defence budget, but were not able to crack it. He is a strong believer you need to work with a specialised health software developer.

Networking

Defence runs its own secure network, which they use. However, this makes it hard to interface with the public healthcare systems, due to a firewall, "Like the Berlin Wall during the Cold War". They run their own server system with redundancies + off-site backup. In theory, they should never go

down, but in practice they do. They act as a mini EHR - an individual's records are accessible at all of their locations.

They will be one of the early adopters of the National Electronic Health Record system. They want to support it early to ensure it moves in the direction they want for the future - and are interested how far it will go.

If they deploy to another country they have two options. One, section out a part of their database, and physically take it with them. Then when they return, they'll merge it back into the main database. Technically, the most robust system, but not that practical.

John prefers to do an online deployment by making a live link back to NZ, usually via a satellite link. There are security issues to work through, but this is the most practical way.

Magic Wand

I've asked every person I've interviewed, "if you had a magic wand, what would you change about your software". I was very impressed with John's answer, and I think it showed his leadership skills.

"There is no magic wand, no silver bullet. There is a lot of competing demands you have to manage and balance off - based off what you want to achieve, and what your workforce needs. Your workforces view is only a small slice of the pie of what you need from the full system.

If I was to ask my Doctors, it would be around the UI, and being able to rapidly get through a consultation without any delays. Everything at their fingertips - no more than 1 click away. Be able to assist them through the process of achieving the right level of documentation. And it needs to provide decision support - without them necessarily having to ask for it. It's all about time and flow. Every-time you take your attention away from the patient, that experience degrades.

If I was to look at it from the view of my non-registered, my medics. They work under a standing orders framework. As soon as they start into the documentation - it's pre-formatted for them. When they get to the diagnosis, there's a pathway and a prescription. What they need is a very templated way of doing things - to be legally compliant.

If I was to look at it from my manager's point of view - they need the ability to see whats going on. This is where tension shows between providers and managers. Providers don't understand information privacy - don't make public what shouldn't be. Two thousand years of tradition there - well the world's moved on. My managers need to have visibility over the healthcare system - both in quality of record, the flows of information, whats the demands are on the system - whats being seen, all sorts of issues. How is the logistic system going, have we transferred from this drug to that drug?

If I look at it from my commander's point of view - all they want to know is - can that person go to fight. They need to find that info without wading through pages of the system.

There is no one silver bullet - there will always be trade-offs. What we're aiming for a 90%+ compromise."

The Future

To manage their healthcare software they have two full-time staff to maintain and improve it, and one consultant who specialises in security. So they have a lot of resources and capacity to improve their system

Predictive Analytics

One thing they're very interested in is applying predictive analytics to their dataset. Their data set isn't truly 'Big Data' - but it's not small. They are actively perusing how their data could be used for predictions and research.

They have a significantly more complete view of their population compared to civilian healthcare systems. They have very regular data points (almost monthly checkups), know exactly what their soldiers ate, how much exercise they did, where they went - they're in a fantastic position to test large scale data analytics.

However, John knows he cannot 100% trust the data. They're a volunteer military - the personnel are there because they want to be. When an individual gives over information, that could have adverse effects for them - at an extreme, it might mean they're unable to deploy. So they have to take all information provided with a grain of salt. The medical staff aren't just there to provide healthcare; they're also there to provide risk assessment for an individual.

Additionally, they cannot be considered Representative of the NZ population. If they produced any research based off that data, they have to be clear it cannot apply to other populations of patients.

Privacy

Privacy is a common concern with any large scale analytics. John finds it fascinating as attitudes change depending on what group you talk to. People in his generation (50's) are terrified of big data. They don't understand the technology, and they don't know they already live in a world with no privacy. John thinks civilian practitioners are working in a model from this generation, which is very outdated - but they don't know the world has moved on.

However, when John tells his younger soldiers about what they're planning to do with analytics, their response is, "What, you mean you're not doing it already?"

It gets even more complicated when you consider the privacy of information generated from this dataset. Some deployments are classified - so the medical records recorded during the deployment could also be classified. Then, if they generate new information from that information, does that mean the new information is also official information? Would they have to comply with an official information request for it?

They have been talking to the Privacy Commission about what the future of health information privacy will look like, and what needs to happen.

24/7 Instrumenting

They are currently testing 24/7 monitoring on a group of soldiers, where they are, how fast they're moving, their heart rate - everything they can record. They're doing this to refine their basic training and reduce injuries.

"In 5 years, I want the entire Defence force instrumented."

They've also been looking at what is possible with EEG instrumenting, however, they're not in a position to deal with the data volumes produced by it. I asked if they had considered genetic testing, and they're avoiding it for privacy issues.

I was very impressed at how much scope the army had to change their systems, and how far they were looking towards the future. John believed this was mainly due to the Defence Force being a goal-driven organisation. Business cases are put forwards, ranked by importance, and it's decided what will be pursued. When that decision is made, the whole organisation will focus on that goal, and make it happen. So they have a lot of resources to implement change, but also have a culture of change and innovation.

Will

Will is working on creating an AI-powered EHR system. He has a background in General Practice and has been designing and building PMS systems for the last five years. He also works with a large PHO, assessing new Health IT tools.

When Will got in contact with me via email, he told me that he too, was working on a "Totally radical PMS system which seeks to redefine how systems of this kind work". Naturally, I was super interested and had some guesses at what it would be, but I never expected what it actually was.

"The system I'm working with is phenomenally clever AI. It writes its own software basically. You tell it in English what you want it to do, and it does it."

It's being built by an international co-operation, and they've invested a "few million" in it so far, so they are very serious about it. To first understand why it's worth this much investment, we need to look at what problem it would solve.

Why are EHR systems hard to build?

Will pointed out that fundamentally EHR systems are not that complicated - their functionality isn't that different to the services that Google can provide very effectively for free. However, he claimed what made them complicated was disagreements in workflow.

"Oh I can tell you why they're hard to build, that's easy - It's because General Practitioners Argue all the time. And they all think their way is the right way to do things, and they won't listen to anybody else."

There are many different ways to record a blood pressure. Some record it as "bp 120/70", or "bp 120:70", or even just "120/70". Different systems have different ways to represent information, and healthcare professionals have their own preferred ways. Some systems have specific places to record blood pressure, in some you just put it in the text notes.

The reality is that both existing software, and medical professionals are inconsistent in how they record healthcare information. The AI system will hopefully take unstructured healthcare information, and structure it. It doesn't matter how you represent a blood pressure - it will recognise it as one, as it recognises context, and store it consistently

The potential impact of a technology like this is huge. It would allow GP's to work in whatever way they choose, while still maintaining data consistency. A large issue in healthcare is 'Interoperability' - basically a measure of how well one system can interface with another. Shifting a health record from one system to another is a huge challenge because both systems will structure the information in different ways. An AI system could provide an automatic way to format a record for a new system. It also would make it much easier to do large-scale statistical analysis of records - as the data is already consistent, and cleaned.

Their vision of the future is appointments where the GP doesn't interact with a computer screen - the software listens with a microphone, understands what the doctor is saying, and writes the notes for them automatically.

Challenges

Of course, building a technology like this has challenges. Apart from the technical issues, convincing the world that it's safe is another thing. However, Will seemed optimistic, claiming their system is truly three generations ahead of anything on the market.

What makes a good EHR system

We also discussed what qualities EHR systems should have, and he had a few points to make.

Centralised Data Repository

EHR systems are defined by a single record which all parties access - a centralised repository of healthcare information. However, this creates two issues - how do you store the data, and how do you access the data. The format which you store the information in needs to be specific and consistent enough that it can fit all use cases. The biggest issue is trusting the person who stores the data - they have a tendency to view it as 'their data'.

Extensibility

Extensibility is necessary for an EHR system as requirements will change over time. The PHO he works for has been wanting to move away from MedTech 32. They assessed many different options, but in the end decided on Profile for Windows. He described profile as 'rubbish' - but it was his second choice, because of how extensible it is. It simply presents the data, in an HTML based shell. In 15 minutes he could start changing the interface and making it fit their needs. It's not complicated to make it feel good, look good.

He thought healthcare systems should be designed as composable components, which provide high-level instructions which can be configured as needed.

Bryan

Bryan a GP at a medium sized community practice, which uses MedTech32. He originally trained and briefly worked as a GP in the UK, but moved to New Zealand 11 years ago.

The unique thing about Bryan is his obsession with process.

"I am crazily obsessed about process. That's why none of my quality markers are missing, everything is filled in it's all done."

Every part of his workflow is incredibly considered and consistently followed. For every single process he does, he tries to shave 5 minutes or 5% of effort off it. This reduces his accumulative workload, and therefore he's always more likely to be ahead of the game, and not catching up on work. "One of my abiding memories were how GPs were always running behind time, I realised in many ways we were just not working efficiently as a profession."

Hardware setup

Bryan uses a large 24" monitor for MedTech32, and a secondary monitor for email, iMessage, browsing the web, and post it notes. For input, he uses a mechanical keyboard with no letters on the keys - he avoids using the mouse as much as possible. He used voice dictation software for a few years, but found that it doesn't do a perfect job, and notes would require proofreading - which was undermining any time saved by using it.

Software setup

Bryan noticed when he first started working in New Zealand, MedTech's tab system. To swap between sections of the patient record, you need to open that specific tab. If you don't open the tab, you'll never know what's in there, or not. He found that many people wouldn't have drug allergies filled in, because they weren't in the exact spot. It was probably noted in the main body of the consultation notes, but not in the exact 'drug allergies' tab.

Viewing this as a downside, Bryan found that MedTech32 would allow you to open all tabs at once, and arrange them as you'd like. He prefers this as it lets him visually see all of a record at a glance, giving him a quick overview.

"I have figured out ways to overcome the problems with MedTech, but that's only because I'm a little bit nuts. Most people live with the problems."

The diagram below is his arrangement, and he explained why this worked best, and how.

Previous consultations		Classifications	Medications		
		Consultation Notes	Warnings	Daily Appointment Schedule	
			Recalls		
Waiting Patients	Screening terms	Lab Results		Outbox	

Consultation Notes

The consultation notes window is where he writes the notes for the patient he is treating. He puts it right at the centre of the monitor, to avoid distraction and stay focused.

He uses a structured template system for his notes - diagnoses, examination, history, plan. He specifically puts diagnosis right at the top of the note, as that will instantly tell him if he needs to read the rest of it when looking through past consultation notes. For example, he would read an annual review of a patient in more detail, than a viral cold. In traditional SOAP notes, the diagnosis (or assessment) is put 2/3rds down the notes, which he considered absolutely daft, and a waste of time.

The other very important part of his template is the plan - he always puts it at the bottom of the note, and lists actions in bullet points. That way it's always absolutely certain what the plan is - what he's done, when he's expecting improvement, and if needed when he'll see them again. He said many GPs would make notes which you can read that they came in with a chest infection, but it's just not clear what was done with them, or what the follow up was, or what safety netting was in place. Bryan believes good documentation is the cornerstone of good practice - for both the GP's protection, the patient's protection, and just good outcomes.

He's been using a formalised system since he came to NZ. He tries to tell a story, instead of abbreviated consultations, so if he reads something 3 years old, he can recall the consultation clearer. It's taken him 9 years to convince the other GPs at his practice to use it, but now everybody uses it by default. A compliment he has received from GP's who have tried his system is that it made them more conscious about what they're writing.

Classifications

The classifications list is where you write down the major points of their medical history. He places it directly above the note as he needs to be constantly aware of their major medical conditions, and he needs to take it into consideration during day to day practice.

One critique he has of MedTech32, is that classifications cannot be ordered. Some classifications are simply more important - knowing a patient has had a stroke, or has heart disease is more important than knowing they had something less critical, such as a hernia repair. By the time a patient is significantly elderly, you might be obscuring the important classifications, with less important ones.

Patient classifications are sometimes called 'the problem list' by some GPs - which causes issues. Some GPs might add things such as, "Can't get out of the house", which isn't a medical condition. There is individuality in practices in how they work.

Medications

Directly next to the classifications, is the patient's medication list. It makes sense that next to their major medical conditions are the medications treating them.

The medication list needs to be broken up into long term, and short term medication. Short term is something prescribed for a one off condition, say antibiotics for a sore throat. Long term medication are medications taken regularly. It's important to separate the list, so it's readable, but also makes it hard to accidentally mess up renewing medication - either renewing something they should not have had, or missing a regular medication. Bryan also thinks a 'renew all' button is critical, instead of doing them one by one. If a patient is on 10 regular medications, that's 40 clicks a year.

Warnings

MedTech provides warnings if you try to prescribe a patient a medication that they're allergic to, or that could interact with existing medication. Bryan has logically placed it beneath the medication window.

I asked him about obtrusive notifications, which alert with a popup. He said that in theory that's nice, but in practice you get patients with 25 notifications, and you end up by reflex just clicking through them - there's just too many to pay attention to. That is actually quite dangerous, as you might miss a really important notification. You're going to have so many reminders as a doctor - from other bodies as a pet project, that notifications really have to be configurable.

Recalls

Recalls are planned future appointments, usually for annual reviews, cervical smears, retinal screening, or follow up after a consultation. This provides an overview for the patient's year.

Outbox

Outbox is where documents are sent to other people on the patient's behalf, such as referrals to specialists.

Lab Results

Lab results are directly below the consultation note - this makes sense as the two most dominate pieces of info (classifications and lab tests) are closest to it. "So if I suddenly see their sodium is 120, I think, I should do something for that"

Screening terms

Screening terms are quick references to things such as blood pressure, or cholesterol readings. These are usually entered in the main notes, but they enter them here as well, as they're faster to find than searching through consultation notes.

Keyboard Shortcuts

In Bryan's own words, if you want to make good medical software, "you need a fuckton of keyboard shortcuts". He avoids using the mouse as much as possible - setting up his own keyboard macros to customise the shortcuts that cannot be changed in MedTech32.

Opinion on EHR systems.

I asked him for his opinion on EHR systems, compared to the EMR system he was using. He noted that he remembered people trying to create working EHR systems when he was a medical student. He could absolutely see why centralised notes would be fantastic, but one of the downsides is notes where anybody can write anything - they end up being filled up with irrelevant information. So as much as it's useful, it can become distracting, hamper the other medical practitioner, and just simply not be safe.

He gave the example, that district nurses started using the screening terms window in MedTech32 to note that they've seen the patient. Which reduced the usefulness of that window for him. It's just not going to be relevant to a respiratory physician to know that the patient had constipation earlier in the year, or for him to know the full details of what a neurologist has been doing with a patient.

There needs to be a system to ensure only relevant information is shown, and that irrelevant information is hidden or quickly summarised.

Survey

As additional user research, I had the opportunity to send an email survey to a GP ICT research group, run by The Royal New Zealand College of General Practitioners. This group was around 85 NZ GPs who had self-selected that they had an interest in Health IT, and out of that, I had 19 responses.

The survey was a short five questions, as I was advised that GP's get many emails. I focused on open-ended questions, about their likes, dislikes, wants, and other thoughts. As I knew the individuals were all GP's, the survey was anonymous (Martin & Hanington, 2012).

Below is a thematic analysis (Guest, MacQueen, & Namey, 2011), or you can view at the end of this chapter.

Common Themes

The single largest theme evident was a latent dissatisfaction with MedTech32, which was used by the large majority of respondents. There are two sub-themes within this.

Slow Performance

Slow performance was a major complaint, and the effect it has on patient outcomes. Web-based plugins and integrations were specifically mentioned.

"1. I work in a large practice (30 gp's) and it takes a long time no less than 6-8 seconds per item to prescribe a medicine when I press the F10 button. 2 Web enabled forms are great but in practice slow down our workflow..."

"Today for instance, I had the plug-in web based applications non functioning. This results in a few minutes of wasted time per time of use. Sometimes more than one incident per consultation. These were ACC forms and lab request forms. Yesterdays problem was the prescribing assistance for Pradaxa hanging for 5 minutes. Many of my colleagues are not prepared to wait this long, crashing out of the patient medical record system and writing prescriptions by hand. Hence the drug with high risk of adverse effect if not properly monitored does not appear on the electronic medical record at all."

"Often...when it runs too slowly or its association with other systems (eg Best Practise, healthpac) make the system run even slower / don't work at all."

"Lots of add ons which slows software"

"it runs too slowly or its association with other systems (eg Best Practise, healthpac) make the system run even slower / don't work at all."

Poor Customer Support

Many were dissatisfied with how the company responds to customer requests.

"When we changed hardware Medtech threw up a lot of error messages; it took a long time with escalation of enquiries at the support desk to realise that there was a very quick fix. The error messages and numbers were of no use to identify this problem"

"I basically hate Medtech their support is appalling..."

"There were drug transcription errors in GP2GP notes transfers which could have had serious consequences for the patient and Medtech were very slow to sort (them) out"

"It is a dinosaur system and the developers have been sitting on their backsides because of a lack of decent competitions"

Simplicity and Familiarity

However, MedTech's simplicity and familiarity was appreciated

"Medtech is slow and old but it is familiar and very reliable."

"Relatively intuitive - not complicated to use."

"It works most of the time. Easy to use basic functionality. However, there is no advance functionality beyond that. The system has not changed significantly since introduction 15 to 20 years ago"

"I know it, and used it for years, it is all simple. Other than hiccups on a single PC the system is stable. Locums and all new doctors and nurses have used it before. But I wouldn't say there is anything really useful about it."

Features to save time

I asked GP's what features they would want to add to their software - a common theme was time-saving features.

"I would love to be able to tick a combination of typically used medicines to facilitate easier / faster prescribing."

"Have the ability to add "auto complete" PDFs by the practice its self"

"More electronic forms rather than having to print out paper ones, integrated best practice guidelines to assist with management"

"speed, format of consultations"

"Make it all one sign on and password, not several. Make it one tab/drop down not multiple entry points."

Features to improve patient outcomes

In addition to that, many suggestions revolved around patient outcomes.

"Addition of learning or evidence topics in the workflow of the EHR and gantt chart of medication use"

"Have a field to show when patient demographics were last updated and a prompt to check if not up to date. Similar prompts would be helpful for long-term medications and "classifications" (problem lists)"

"...there is scope for a lot more automation with incorporation of clinical guidelines into a PMR to ensure patients' results are appropriately actioned and that treatment plans meet current gold standards (I believe our PMRs should even out variability in practice between GPs so that every patient can achieve the same clinical outcomes)."

Modernisation

There was a strong call to modernise MedTech - both aesthetically and internally.

"Aesthetically MedTech as a system looks old and outdated - even MedTech Evolution doesn't look like a modern system."

"Make it more modern rather than working on a 1990 frame with things being added on"

"Medtech - I would change it so that it was less Dos and more Windows"

(In response to If you could make an update to your software, what would you change?)

"Almost everything."

Arguments

In this section, I argue 3 fundamental challenges faced when building healthcare software.

1. Healthcare is complex, and varied.

The first challenge of building healthcare software, is the complexity of modern care. For example, Practice Management Systems, (such as MedTech 32), can include the following features:

Patient Records	Storing a complex record of patient history
Appointment Management	Scheduling patient appointments
Invoicing	Generating and sending invoices to patients
Claims	Claiming subsidisations from the Government for medical services provided.
Prescriptions	Creating and verifying that prescriptions are safe
Patient Population Analysis	"How many of our patients are under 30 and smoke?"
Decision Support	Providing references and resources about specific conditions when needed.
Staff Scheduling	What shifts are our staff working?

To ensure these features work, it has to integrate with many institutions - this answer from my survey of New Zealand GP's who are interested in IT said it best:

"As GP practice is so complex with its interactions with so many organisations the PMS has to be able to interact with DHBs, Private organisations, Insurance companies, ACC, PHO, MOH (Immunisations, GMS), NHI look up, eSA, eSAM, GP2GP transfers etc"

The scope of functionality and responsibilities a PMS system has is huge. In the world of software development, software like this can be considered to have a 'monolith structure'.

This has the effect that it leads to strange dependencies. For example, you could argue that appointment management and patient records should be handled by the same system. The doctor should be able to view their next appointment, click on it, and automatically open up that patient's file. They should be able to view when the patient next has a scheduled appointment. Likewise, the appointment management tool needs to get a list of current patients,

With that same logic, you could argue that the tools used to analyse your patient population (e.g., "How many cases of the flu have we seen this winter?", "How many patients over 30 are smokers?"), should be connected to your patient records. It makes sense as the analysis tool needs to get its data from the patient record database.

However, this means that your analysis tool is dependent upon your appointment management system. Say if you made a change to your appointment management system, which caused the program to crash. Now you can't use the analysis tool until the bug is fixed.

The other downside of this monolith structure is that it makes it harder to specialise. Let's say you had a fantastic idea on how to improve the patient appointment process - your software could reduce missed appointments by 10%, and the office staff will find it a joy to use. If you went and built this system, it couldn't be easily swapped into their existing system, as a replacement for their previous appointment management tool. If you want to get a clinic to use your system, you'd also have to build all the other parts of a patient management system - patient records, invoicing, prescription management - all of it.

Variety

Modern healthcare is complex. However, what makes this even worse, is the variety of healthcare.

Healthcare is surprisingly regional. Earlier I talked about a variety of organisations that a PMS systems have to interact with in New Zealand. If you built that product and tried to bring it to another country - you'd quickly find that they have an entirely different set of organisations to interact with.

Even within nations, such as New Zealand there exists a lot of regional variety. We have a number of 'shared care' record systems. The start of an EHR system - services which store a simplified medical record, which can be accessible at authorised medical locations, such as after-hour clinics, hospitals, and paramedics. Within New Zealand, we have three separate systems - SharedCareRecord, Whānau Tahi and HealthOne.

Variety also exists within users. Karsh et al (2010), state that a user's needs of their software can change tremendously between:

- Clinical Roles (nurse vs. physician)
- Clinical Situation (acute vs. chronic care)
- Clinical Environments (intensive care unit vs. ambulatory clinic, etc.)
- Institutions (Auckland Hospital vs. Wellington Hospital)

Different medical professionals vary in the information they need to consume and store. For example, your specialist treating your eczema will be interested in very different information, to your physiotherapist treating your cycling injury.

Even when two parties are interested in the same information, they can need to view and interact with it in very different ways. When your pharmacist views your medication list, they want to find out how much medication they can give to you. However, when a paramedic looks at a medication list, they want to know if you've recently taken any medication which could affect the emergency situation. This requires completely different user interfaces, with completely different interactions.

2. Risk

It seems obvious, but building software to manage health information has a huge risk factor. If information is lost, incorrectly recorded, it can have terrible real world effects.

In 2015, an elderly man went to his GP with ankle pain (stuff.co.nz, 2015). He was prescribed the painkiller Voltaren, and advised to return in a months time.

However, this patient was allergic to this medication. His medical record did show this; it noted to avoid the medication as it previously caused problems with his renal function.

Usually, the GP's software would show an alert if he tried to prescribe a conflicting medication. However, their practice was merging with another medical centre at the time, which was possibly causing computer difficulties. The GP stated he never saw any alert or warning.

The patient returned in a month, with pain in the joints of his right foot. He was diagnosed with probable gout, and advised to keep taking Voltaren.

Two days later he was admitted to hospital and diagnosed with renal failure. He passed away.

So who's at blame? The commissioner said the GP failed to provide services with reasonable care and skill. He was also critical that the practice didn't ensure its systems were fully functioning while the practices were merging.

However, the fact is still that the software failed, and could've prevented this death.

This tragic story shows the challenge of managing risk in healthcare, and the many ways failure can occur. Failure can include:

- Software Failure
- Hardware Failure
- Human Action (accidental or malicious)

Software development has spent a lot of effort mitigating the first two. In critical areas such as banking or large scale data centres, software is tested rigorously, and designed to deal with hardware failure.

Human action can be accidental, such as a software developer accidentally introducing a bug in a software patch, or may be malicious. I interviewed Micheal, the founder of a healthcare software company - and he told me a scary story. Their developer locked them out of their servers, while the service was running and demanded more payment - putting a population of patients at risk. Legal action was taken, but initially they had to pay the money to continue access. In his experience, he found that human action was one of their biggest risks. The other largest risk for them was a financial failure - without a company to support their software & service, it wouldn't be able to function.

Institutions are also a potential target of attack. A Hollywood hospital was hit with a ransomware attack (Winton, 2016) - this is a malicious piece of software which encrypts all of their computer's files. The institution decided the fastest way to restore system functionality was to pay the \$17k ransom (paid in the form of bitcoins), which they did.

They were forced to return to pen and paper while the system was down. This is not all uncommon - Winton reports that according to federal records, between 2010 & 2016, at least 158 medical institutions have reported being hacked or having issues that compromised patient records in the US.

Two months before this thesis was published, in May 2017, UK hospitals were hit with another ransomware attack (Brandom, 2017). This attack caused 16 hospitals to have to be shut down. The virus was not specifically designed for hospitals but spread rapidly there due to the use of out of date IT systems.

Impact

Developing software for healthcare inherently includes more risk than many other types of software. Loosing medical information can be disastrous, loosing a high score on a mobile game - not as impactful.

Risk must be considered and managed in every development decision - to protect both institutions, patients, and the developers. However, I do not believe risk should be a reason to impede innovation in healthcare software - discussed more in the next section.

3. Healthcare software is hard to distribute

Even if you've built a fantastic piece of software, implementing it in the real world, is a huge challenge. I realised when interviewing GPs, that actually getting your product being used by a GP, for their day to day work is a huge challenge. This is for several reasons.

Selling to institutions, not users.

Healthcare software is distributed to entire institutions, not individuals. That means that if one doctor wants to use a different piece of software, every doctor needs to make the switch as well. The larger the organisation, the harder it is to find software which everybody is happy with. This makes it really hard to get early adopters - you're not trying to find beta users, you're trying to find beta institutions. One GP I interviewed, Paul, was a beta tester. He was in a good position to do that, as he was the sole GP at his practice - if he wanted to try a new piece of software, he only needed to make the change for himself.

Switching software is risky for an organisation.

Changing any system is a risk for a healthcare organisation, but switching out a piece of software as complicated as a patient management system, is a huge risk. There is the risk of patient records being incorrectly converted from the old system to the new. There is the risk they will make mistakes early on. What if the new system fails?

Switching software is costly for an organisation.

Where there are risks, there is cost. You will need to pay for patient record conversion and staff training, which is usually provided by the vendor. The new system could make your workers inefficient while they get used to it. The institution may have to pay the vendor or IT staff to help install, configure, and test their new system.

On top of the external costs, institutions also need to spend employee time to find and assess different vendors systems. The institution needs to feel confident that the new system will fit their requirements, and be an improvement over their old system. The question an institution has to consider is, "Will the upsides of new software outweigh the cost, and risk?".

Socio-technical Change

A well-discussed concept to consider is the socio-technical aspect of healthcare software (Coiera, 2004) (Beasley, Holden, & Sullivan, 2011).

The reality is that technical systems have social consequences. The way software works will affect how users, and institutions as a whole, use them. Likewise, technical systems are built based on social systems - how an organisation works will affect how a tool is built to that need.

Coiera concludes that we don't build technical systems - we build socio-technical systems. This requires an intimate understanding of how people and technology interact. They simply cannot be designed independently.

This can be considered as another explanation as to why healthcare software is hard to distribute. If a new piece of software has an effect on the existing social systems of an institution, it will be hard to distribute that software.

"Accelerating Innovation in Health IT", also touches on the topic of distribution (Rudin, Bates, & MacRae, 2016). Common methodologies in both design and software development use an iterative cycle of building and testing. This requires both the opportunity to fail, and fast distribution.

"Developers can serve users' needs better when they have the freedom to experiment and fail quickly."

They suggest the creation of 'sandboxes', environments where it's easy to distribute new versions of software, and where failure doesn't have disastrous effects. This is one approach to reduce the challenges of risk and distribution during the development phase.

The Value created by new tools.

It's worth considering the value created in a healthcare system by better software. A doctor in NZ with the current industry standard software will see between 24-32 patients a day, depending on appointment length. Would a doctor with improved software be able to see more patients per day? While better software can certainly create more efficient workflows, reduce mistakes, and provide better patient outcomes - how much effect would it really have on the bottom line?

The Interaction between Complexity, Risk, and Distribution

The three reasons I've argued don't exist independent of each other. Arguably, the difficulties of distribution are actually caused by risk, and complexity.

The more complex an existing software system is, the harder it is to create a replacement - as it's workflow is ingrained in the institution. You can't just replace a portion of it, you have to replace the entire system. So the new software has to match all of the existing functionality.

The new system must provide more value than the existing system to make it worth upgrading - usually, this means more functionality. This creates an even more complex system. As the complexity of these systems grows, the risk of replacement increases.

Ultimately, it comes back to the question of potential value versus the risk + cost of upgrading.

Brief

Originally, when I started my thesis, I was planning my output would be to create a better version of the software my GP uses (which was as I discovered, an EMR system). However, during this research phase I found problems in Health IT that I could not ignore.

To create an EMR system for the NZ market would be both out of the scope of my thesis (it'd take a team of people, years of time, and a lot of money), but most importantly, it would do nothing to tackle the underlying problems I discovered in my research. A new product would have little impact on NZ's Health IT, as the challenges of Complexity, Risk, and Distribution would remain.

Instead, I've decided to take my output in a unique direction. I'm going to create a design proposal for a unique, experimental EHR system. I have some clear goals I want to focus on:

Be experimental

There is no value in making a design proposal of something which is similar to existing software. The opportunity is to explore and propose ideas which potentially have a high impact on Health IT. I'm going to be focusing on new technology, and how existing technology could be leveraged differently to design an EHR system.

Do one thing well - manage health records.

My output is going to focus entirely on the problem of managing healthcare records. This is how are they stored, shared, and interacted with - and nothing else. This means the scope of its functionality is quite limited, for example it won't do things such as appointment management, staff scheduling, invoicing, etc.

This is for a few reasons. Firstly, as complexity grows, so does cost and risk. Secondly, this aims to make distribution easier - if an institution needs to replace my system in the future, it has a much smaller scope of functionality, and therefore is easier to replace. And thirdly, because it aims to be general purpose.

The value of software with a scoped focus has been long understood in the world of software development. In the original UNIX philosophy from 1978, the first rule was "Make each program do one thing well." (McIlroy, 1978). This rule has been long celebrated and has been re-worded into every subsequent UNIX philosophy.

Be a general purpose system - usable across institutions.

Healthcare software is currently designed and built around the needs of institutions. If my output aims to be focused on the general problem of health record management - then it should be usable across institutions.

Respond to the larger issues.

In every design decision, I will try to tackle the larger issues I discovered in my research - The challenges created by Complexity, Risk, and Distribution. My output is a response to these issues.

It's been well discussed that the issues with distribution impede innovation (Coiera, 2004) (Rudin, Bates, & MacRae, 2016) (Beasley, Holden, & Sullivan, 2011). How can my system be designed to enable faster distribution, with less socio-technical impact? How can my system reduce the inherent risk of healthcare software? And how can complexity and variety be managed?

Design Methodology

An EHR system is a complicated piece of software to design. Instead of tackling it as a whole, I've decided to split the problem up into 4 separate areas:

Storage Format - How do you represent healthcare information in an EHR system?

Authentication & Privacy - How do you verify that somebody is allowed access to a record - and how do you protect the patient's privacy?

Networking - How will the record be transmitted and shared between all parties?

User Interface - How does the end user interact with this system - and how would software interact with this system? Dissatisfaction with existing interfaces was a common theme discovered from my user research.

While there are interactions between these areas, they clearly divide the problem up. This way I can conceptualise and iterate upon smaller problems faster, select the most successful outcomes, then combine them to create the final system.

Criteria for assessing concepts

When assessing how successful concepts are, I'm looking at three separate qualities, which relate to the goals discussed in the brief:

Potential impact - Could this idea have a large impact on NZ health IT, and more specifically, how could it address the three issues of Complexity, Risk, and Distribution?

Feasibility - Is this concept reasonably achievable? I don't want to create a design proposal for a system which wouldn't be able to be built. This project is not a speculative design project (Auger, 2013) - all concepts must have firm evidence of feasibility.

Uniqueness - While I will look at existing solutions to these problems, there is opportunity and value to assess unique approaches.

Storage Format

An EHR system's main purpose is to store and update health records. So if you want to build an EHR system, your first thought will probably be - what exactly is recorded in a health record?

I was quite surprised to find out there is no standard definition for a health record. Most people I talk to outside of the medical field are also quite surprised by this fact. Unlike an audio file, image file, or text file - there are no industry standard file formats for medical records.

Specifications

So what information should be stored in a medical record? Some of it is expected. For example, here is some generic information commonly found in New Zealand primary care records.

Demographic & Admin Info	Name, age, NHI Number, ethnicity, address, email, phone, etc
Medication list	Brand name, generic name, dosage, purpose
Allergies	Allergen/drug interaction, severity, reaction
Clinical Documentation	Purpose of visit, typed notes, treatment given, attached prescriptions
Lab Results	Copy of results from lab, details of sample
Immunisations	Name, date, reaction

However, this is just a medical record - not a health record. If a surgeon tried to use this record, they wouldn't know where to put their procedure details, or a physiotherapist wouldn't know where to track their patient's muscle strength. The information a GP needs is very different to what a hospital needs, to what a physiotherapist needs. All of these medical institutions deal with very different information, so all of their records are, by nature, very different.

Karsh et al (2010), state that a user's needs of their software can change tremendously between:

- Clinical Roles (nurse vs. physician)
- Clinical Situation (acute vs. chronic care)
- Clinical Environments (intensive care unit vs. ambulatory clinic, etc.)
- Institutions (Auckland Hospital vs. Wellington Hospital)

They conclude that,

"To succeed in today's team-based healthcare reality, health information technology should ... recognize that each member of the collaborative team may have different mental models and information needs, and support both individual and team care needs across multiple diverse care environments and contexts."

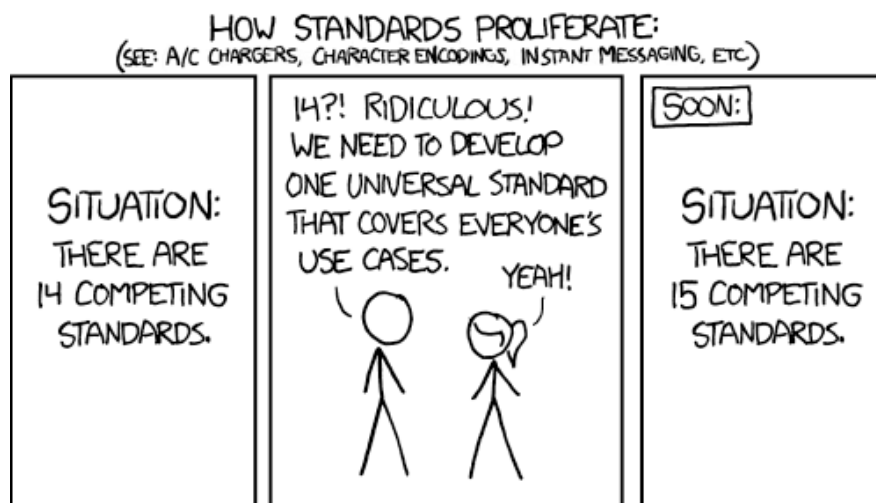
For this project, I have decided on my own definition of a health record.

A health record contains any information relevant to the patient's health - as defined by their medical staff and themselves.

This means that I cannot exactly define what information will be stored, or could be stored in the future, and that must be considered in the storage format.

Concepts

1. Existing Format



As this xkcd comic points out (Randall, n.d.), perhaps it would be better to support an existing standard than creating a new one. The most promising standard I've found is FHIR (Fast Healthcare Interoperability Resources), from the HL7 organisation (Health Level Seven International).

FHIR provides a standard format for most healthcare record cases. It is by design, an extensible system - it is designed to be adapted for local usage, so a FHIR record (or resource as they call them) from France would be different to one from Auckland.

FHIR is more than just a record format however - it also provides standard REST apis for sharing medical records through HTTP. FHIR however, does not handle securing or authenticating the api, that is up to the implementor.

However, FHIR is not designed as a storage format - it's designed as a common format to help reduce the issue of interoperability. For example, say you wanted to move records between system A and B, which both use different record formats. First, you would convert the A records into FHIR, and then from FHIR into B records.



2. Blockchain

Peer to peer technologies have provided some of the most innovative data storage systems since the creation of the Internet. Blockchain technology has had the biggest impact in this field since the original BitTorrent protocol. The blockchain is a distributed database most famous for powering the crypto currency BitCoin. It's cryptographically secure, and able to scale to worldwide usage while being independent of any government or governing body.

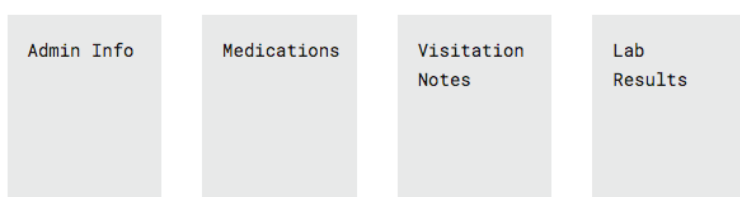
So why not store health records in the blockchain? Simply because the blockchain itself is not designed for large scale storage. It is essentially a shared ledger - it works because every single user can keep a whole copy of the ledger for verification. This is why it's best suited for things which don't take up much data, such as transactions - as of writing, the bitcoin blockchain contains over 180 million transactions, but that only takes up 97gb.

There has been projects attempting to store medical records on the blockchain (Ariel Ekblaw, 2016) - however, in reality they're using the blockchain as an authentication system, which sits above existing storage systems. I discuss this idea further in Authentication.

3. Modular Records

If a health record is essentially a collection of medical records, why not store it as just that. Instead of trying to define a singular record, define a health record as a collection of separate modules, which work together to build a record.

For example, a very simple primary care record for our patient Alice could be made up of the following modules



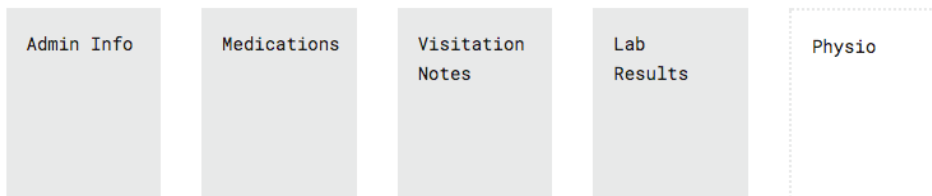
Each module has a defined format (a Schema for programmers). For example, the admin module contains general information needed for patient administration, and could look like this:

The main idea behind the modular format is that it would be Extensible. Say for example Alice's record is opened by her physiotherapist. There isn't a good place in her current record to track her rehabilitation. Or say a surgeon wants to write procedure notes about a surgery - there isn't a place for past procedures. Say Alice wants to put her self-tracked diabetes data into her record - there isn't

```
Admin Info
Name: Alice Smith
Born: 13/1/1983
NHI: AAA0000
Contact:
- Phone: 020 000 0000
- Email: alice@gmail.com
Address:
- City: Wellington
...
```

a module in her record for it, so why can't she add her own?

If users could define their own modules, then there is no limit to what information could be stored in a health record. If a patient, or any health professional around them thinks that some piece of information is worth storing, they could add it. However, I'm not suggesting every doctor starts writing their own formats. The idea with modules is that they are community run - if you define a module, you put it online in a registry, for other people to use. Chances are, Alice's Physiotherapist could just find a rehabilitation module that he likes on the registry, and add that to her record.



There are precedents of a community run module system working well in the context of software development. "Package Managers" are community run systems for the sharing of modular pieces of code. Say you wanted to write a piece of software, and it needed to format a date value as something easier to read, such as "three weeks ago". You could write your own system to format the dates, or you could find a module that somebody else has already made to do it for you.

Node Package Manager (npm, Inc, 2014), is a package manager for the programming environment Node.js. As of writing it has about 370 thousand modules, and the most popular module is downloaded about 1.6 million times a day. Other than illustrating the success of these systems, it provides a sense of quality. If a module has been used that much, it must be dependable and error free.

Within the context of a health record, modules would work in a very similar way. They'd be community run, and community improved. However, instead of being small programs, they'd be templates for sections of a health record.

Version Control

There is one other key features our format must support - Version Control.

There are very few good reasons to actually delete information from a health record. If there is information to be removed, it should be able, but the information shouldn't actually be deleted from the file. Information which might appear to be irrelevant at the time may become relevant in the future. Incorrect information should be able to be updated, but a history of what that information used to be should be kept. There is the argument that inappropriate comments should be deleted - but how can a file tell between removing a snide comment versus covering up information of medical malpractice?

Version Control system have been in successful use for years in other fields. For example Git (Torvalds, 2005) provides version control and team collaboration for code, which is used extensively in software development.

Network

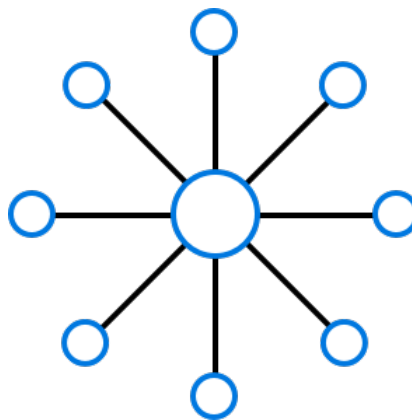
The network of an EHR system has two main purposes:

Availability - Ensuring that healthcare staff have access to the right data when they need it.

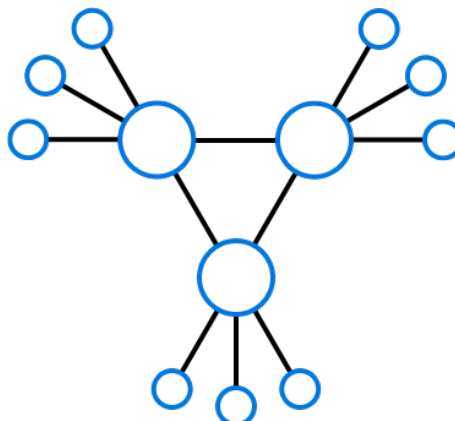
Robustness - Allowing the network to recover from loss of functionality, to maintain availability without data loss.

Networks come in three basic types, as defined by Alexander Galloway, in his book Protocol (2004).

Centralised



Centralised networks are the most basic type of network, a collection of nodes connected to a central node, or hub. A common example is radio - a single station broadcasts to many clients. The central hub holds power over all connected clients, and is a weak point of failure. Most existing EMR systems in New Zealand work like this - a single computer will be set up as a server in the doctor's office, and the doctor's computers will connect to it to share the records.



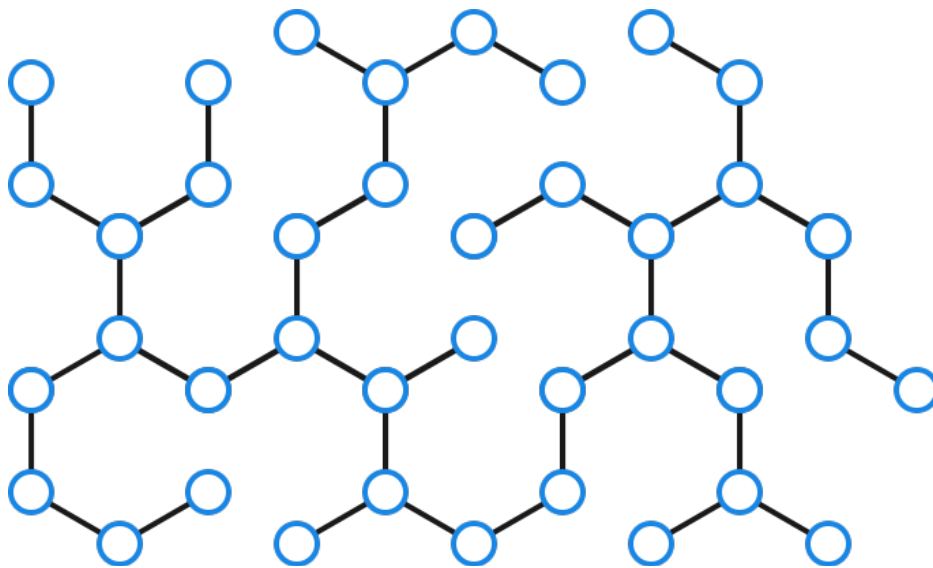
Decentralised

Decentralised networks are essentially a duplication of centralised networks - there is still central nodes, they are just duplicated. The benefit is that if the central node fails, there are others to take its place. Alexander argues control exists in these networks through bureaucracy. The central nodes have to be run by a small group of people, and they have the ultimate say of what the network is used for.

There are decentralised networks used in New Zealand for healthcare, such as the several 'Shared Care' record systems within New Zealand - SharedCareRecord, Whānau Tahi and HealthOne (Compass Health, 2016) (Whānau Tahi, 2014) (HealthOne, 2016).

These services are a service allow part of your medical record to be uploaded to a central cloud platform. The record can be accessed in specific cases, for example if you visit a different GP, if you're in an emergency department / after-hours practice, or in an ambulatory setting by paramedics.

Distributed



Distributed networking is the type I'm most interested in. In a distributed network, there is no defined hierarchy - any node can connect to any other node, or not. No node holds any inherent power over any other, and the nodes can re-arrange themselves free of restriction. The distributed networks of our time have been the most transformative - the Internet, is a distributed network. So are peer to peer networks, such as the BitTorrent protocol, and also crypto-currencies, such as BitCoin. No one company or country can hold power over all of any of these networks.

Galloway however, argues that despite distributed networks being the freest network type, power is still controlled by the protocol itself. In order for a node to join a distributed network, it must play by the network's rules, which are defined in its protocol. Therefore, power still exists, and it is held by the creators of the protocol.

It has been concluded that it is feasible for a distributed healthcare network to scale to provide planet-wide care (Kakouros, 2013). The socio-economic feasibility has not been demonstrated yet; there are very few precedents of using distributed networks in healthcare.

National Health Index

In order for any network to transfer information, there must be a standard way of identifying resources. In the case of an EHR system, the issue is if two nodes both have records for Alice Smith, how can we know if they're different people, or if they're the same person? Thankfully, the Ministry of Health thought of this. In New Zealand, all patients have a National Health Index (NHI), which is a number uniquely identifying them, across providers. This provides a perfect way to identify records.

Concepts

1. Distributed Storage

One unique network strategy has been used by 'cloud' storage provider, Storj (Storj Labs Inc., 2014). They provide a distributed cloud, powered by their user's computers. Files are encrypted, broken into pieces, then spread to other user's computers where they're stored. When a user wants to access a file, the file is downloaded from multiple users at once, which provides fast download speeds. The system has built-in redundancy - the file is stored multiple times in the network, to avoid malicious users.

This way your data is stored in the 'cloud' - but the cloud is created by the user's themselves.

Networks like this do have a potential to scale to huge size - they could technically support the storage requirements for a population's medical records. However, there is one issue - the encryption keys. Health records are a shared record - multiple parties need access to the one record at the same time, and they need to be added/removed as needed. In order to distribute the keys there needs to be a central point where all users access. This ultimately undermines the design of the decentralised system.

2. Fully Cloud Based

Software as a Service (SAAS) products have had an explosion in popularity since the availability of cheap, high-quality cloud computing services. Unlike software which you download to your computer then use, SAAS is used through a web browser. For example, New Zealand's own Xero is a SAAS product.

There is a lot of value in a fully cloud-based system. The first one is lowered IT costs. Software doesn't have to be installed and configured correctly, all your user needs to do is visit the website. A practice doesn't have to purchase a computer to be their practice's server. You as a user don't

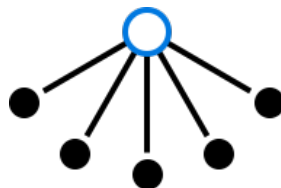
have to worry about backing up your data, as your provider should be doing that for you. And, it can be accessed anywhere a computer can get an Internet connection.

Within the context of primary care, a SAAS model can work. Healthcare companies such as Indici and Practice Fusion are betting on it (Indici, n.d.) (Practice Fusion, Inc., 2014). Their practice management systems are entirely online - they are fully cloud based. The main downside with systems like these is that if the Internet is down - your doctor can't use their software.

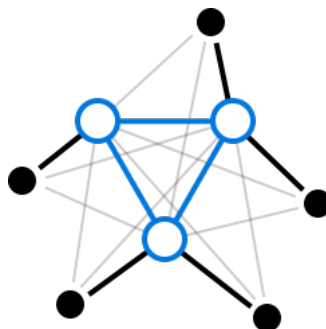
How often this occurs in New Zealand cities is debatable, particularly with the ability to use mobile Internet as a backup. However, looking at the context of designing a system to potentially be used outside of New Zealand primary care - it's not feasible. Medical professionals have to access and use patient data in situations where there is no network connectivity - or where ad-hoc networks have to be created separate to the Internet.

1. Distributed clusters.

Instead, let's first focus on building a network for a singular institution, for example a doctor's practice. Lets say that a single node represents a server, which stores the practice's medical records. The node stores the user's login details, so they know who should have access to what record. This means that doctors (black dots) can login to the node to edit their patient's records.



Now, the bare minimum of duplication required to provide decent accessibility to data is two. One node physically running the pharmacy, and one running in the cloud. This means doctors can access records anywhere the Internet is available, or if the Internet is down in the practice they can still use them. However, if the cloud node has to be taken offline for maintenance, availability is highly affected. Data loss is also not likely with two nodes, but medical records are legally protected documents - losing them would have ramifications. Therefore, I think the minimum optimal number of nodes a clinic should run is three, one physically, and two in the cloud, on different providers.

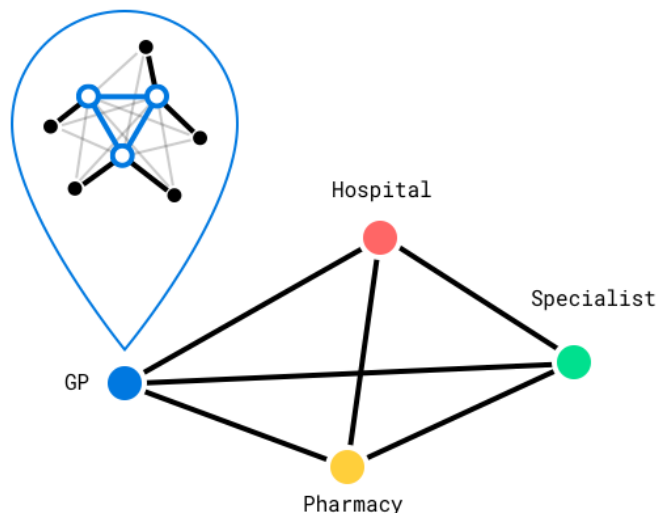


The nodes are duplications of each other - they all contain a complete copy of the records. This is called a 'Cluster'. If a record is changed in one node, it will be synced to the others when next available. This creates an eventually-consistent system, as in; eventually all nodes will have the

same copy of the information. The clients (doctors) can access whatever node they like - and the other nodes provide backup connections (light grey). This way, if a node goes offline, is slow, or can't be accessed; they can switch to another node.

This provides a better system than traditional backups. The nodes don't sync on a schedule (eg once a day at midnight), changes are replicated throughout your network as they are made.

This is essentially a decentralised network. However, this is within the context of a singular health institution. What happens when our patient's records need to be accessed at the hospital?



In this model, clusters are run by institutions (represented by fully coloured dots). Their clusters run independent of any other cluster - so they have full control over how they want to set up their nodes, and how they want to use them.

But then, clusters are connected together to form a distributed network, allowing institutions to collaborate on records. The key is that clusters will only share information with another institution, if it's relevant to that institution.

For example, our patient Alice visits the hospital, for an ankle injury. They prescribe her some medication and note it in the hospital's records. The hospital's cluster notices that Alice has updated her medication - and that other connected clusters also store Alice's medication information. So it sends the latest version of her medication list to her GP, her specialist, and her pharmacist. Now all institutions relevant to her care have the latest version of her information.

After her hospital visit, she goes to a follow up at her GP. Her GP makes notes in her visitation notes. The GP cluster notices the change - but no other connected cluster stores a copy of the visitation notes, so it doesn't update anybody else. In the same appointment she notes that her Eczema has been worse for the last few weeks. Her GP makes a note in the Dermatology section of her record. The GP's cluster sends this to her specialist (a dermatologist) but doesn't send it to her pharmacy, or the hospital - as that information isn't relevant to them.

I think this model works well for two reasons:

Independence. Despite institutions being connected in a network, they are not dependent on each other for service. If an institution is cut off from all of its connected institutions, it still has a full copy of the patient records it requires. The institution can continue to run independently, with full functionality.

Flexible robustness. This model can be as robust as needed. While this model could be run off just a single server, institutions can simply add more nodes to their cluster to increase availability, and robustness.

There is a caveat, more servers are potentially required overall, compared to a SAAS model.

Authentication & Privacy

Authentication has two main purposes - verifying the identity of the user and protecting the patient's privacy.

Authentication Strategies

There are many different ways to verify the identity of a user. Here is a quick overview of some of the most common methods (Pandya, Narayan, Thakkar, Madhekar, & Thakare, 2015), (Kamarudin, Yussoff, & Hashim, 2016).

Email + Password	Common method, used by services such as Facebook, Twitter and Google.
Multi-factor Authentication	Multi-factor checks for several separate pieces of evidence of identity before allowing access, ensuring a much more secure login. Typically, two things are checked - knowledge (such as a password), and possession (such as a cell phone). A common example are ATM machines - you need to know both your PIN, and physically have your bank card with you.
Passwordless	Passwordless logins will email or text you a temporary, unique password which expires in a short time frame.
Physical 'key'	Physical objects can be used to verify identity - an example are key cards, which are commonly used as locks on doors (such as in a hotel).
Biometric	Biometric authentication works by looking at a unique physical feature of the user - such as a fingerprint, face, or iris.

The question of 'what method is best for an EHR system', is tricky, as EHR systems can be used in so many different contexts.

For a GP logging into her computer every morning, multi-factor authentication would be easy, and secure. However, imagine a nurse in a hospital, who has to switch between many different computers as he moves through the ward - it would take forever. In that case what would be the best would be a physical keycard, which would enable him to log into any machine with one swipe.

And imagine the patient trying to log into a patient portal to view their records - they might not do this very often. It would be easy for a password to get forgotten and lost among the countless digital services we use in our modern lives. What might be much easier for them would be a passwordless system, where a temporary login link is emailed to them, or a code sent to their phone.

In order for an authentication system to cater to the range of use cases in Healthcare, it needs to be designed with multiple strategies in mind. By allowing institutions to use what authentication strategy makes sense to them, ensures they can balance security, and ease of use for their staff.

MedRec

MedRec is a very interesting project, which uses Blockchain technology (which powers Bitcoin), as an authentication layer for health records. While this project certainly fits my requirements for being experimental, I don't think it's appropriate for my project, for the following reasons.

MedRec is an authentication layer, which sits on top of provider databases. So a patient's health information isn't stored in the MedRec Blockchain - it only contains the authentication information. While this ensures the authentication is cryptographically secure, there is no guarantee the provider DB is secure. They do recognise this as a caveat, "MedRec does not claim to address the security of individual provider databases where the record content is stored. This must still be managed by the local IT admin".

The value MedRec provides to patients is a single authentication identity which can work across multiple institutions and providers. However, what value does it provide to institutions and software providers? Medical researchers are provided with value by gaining access to anonymised metadata in exchange for computational resources (mining) which sustains the network. However institutions and software providers - the stakeholders who have to make the largest changes, aren't provided with any strong value.

Institutions would have to make changes to their workflows due to the new software - and software providers would have to re-implement their authentication layers. Both are non-trivial tasks.

MedRec is really interesting research, and I hope research in this area continues - but for my project, I believe it adds too much complexity, without creating enough value.

Filtering

It's also worth considering how information in a record can be 'filtered', depending on the user. This is for two reasons:

Privacy. Healthcare information is inherently sensitive information for patients.

There are varying levels of sensitivity - for example, a patient's medication list should only be shared with medical professionals who are caring for that patient, but all parties should have the most up to date list possible.

But at another extreme, some information is shared with only a single medical professional, in complete confidence. This is because the information could be embarrassing, have a huge mental effect on the patient, or otherwise.

Relevance. As a GP I interviewed noted, Bryan, not all information is relevant. He could understand the value of EHR systems, but he was also concerned with his records being filled with information that would mostly get in his way.

He gave the example that while it would be useful for him to know his patient visited a neurologist - the full details just aren't relevant to him, and would have no affect on how he treats his patients.

How filtering works will depend a lot on how the information is structured. If the data was structured into 'modules', as I suggested previously, it could make sense to filter on a module to module basis.

Access

As much as authentication is about ensuring the incorrect people can't access records, it's also about allowing access to those who need it. So far this chapter has mostly been about authenticating medical professionals - but what about the patient?

As a personal experiment, I requested a full copy of my records from my GP. The process was quite involved. First, my GP had to approve the release of my records, so I discussed it with him at a regular checkup. He was quite happy to hand over my record, but just wanted to discuss why first. Then after a few weeks, I received an email containing a locked ZIP file. After replying to the email that I was indeed Eliot Slevin, they sent over the password. Inside the ZIP was a PDF file of my entire record. It was formatted to be printed on 86 A4 pages.

Reading my own medical record was a strange experience, as my visitation notes dated back to my birth - 22 years. It was a bizarre trip down memory lane of my sicknesses, accidents, and otherwise. However, it was completely fascinating, and I highly suggest to anybody who's curious about reading theirs, to do it.

Accessing my record like this had its drawbacks. Firstly it took an extended period of time, and secondly the format of a PDF isn't that useful. It would be hard to take that data and use in a software program, visualise it, or analyse it.

Patient Portals

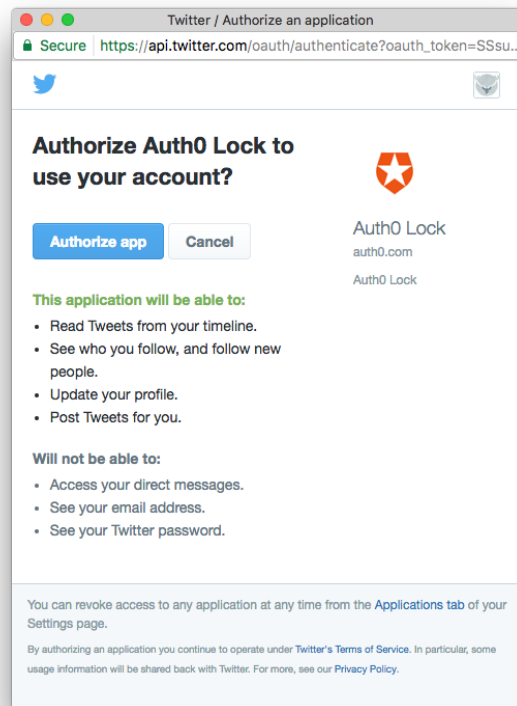
Patient portals are a great way for patients to access their records. The patient's functionality can include viewing their record, but can be extended to booking appointments, requesting repeat prescriptions, and even messaging their GP.

A Nurse Manager I interviewed, Olivia has found great success with their patient portal. It's very popular, with 70% of their adult patient population is signed up to use the patient portal. A common thought is that digital solutions such as patient portals aren't used by the elderly - but there is certainly demand. The first patient they signed up was 80 years old, and he's still using it.

Patient portals are great for patient accessibility to their record, and their healthcare provider, but they don't do much in the way of allowing data access in a programmatic way. In order for me to extract data from a patient portal, I'd still have to manually copy the data out.

OAuth Login

The OAuth 2.0 protocol (Hardt, 2012), enables a third party limited data access. It's currently used by some of the largest Internet services such as Facebook, Twitter, and Google. For example, here is a screenshot I took of Twitter's current OAuth dialog.



OAuth would be a good choice for a protocol to allow 3rd party access, for several reasons.

Familiarity. OAuth is used by many different services, and it's a system which many patients are already familiar with.

Authorisation rules. OAuth doesn't just allow you to authorise a service to use your data - it can also be used to customise what aspects of the data they have access to. For example, Twitter explicitly says what information the application will be able to access, and what it cannot. This flow will ensure that it's explicit what information they're authorising access.

There are many cases where this would be appropriate. For example, in many aspects of care medical professionals will ask patient's to keep a 'diary' of their condition. For example, a headache specialist could ask a patient to track details of their migraines, so they can see what factors have an effect. A smartphone app could be designed specifically for tracking migraines, and allow the information to be automatically written to their official medical record. Physiotherapists could use apps to allow patients to correctly track their muscle recovery after injury. Perhaps one day, machine learning services could be created to instantly give you insight and feedback into your health based off of your medical record.

Platforms to build 3rd party healthcare software already exists, such as HealthKit (Apple Inc., 2014). Healthkit provides a platform for app developers to track, and store health information on Apple Devices, such as the iPhone. Healthkit can be used to track sleep, exercise, heart rate, and any other type of data. But, what Healthkit can't do, is add that information to patient's official health records.

I have personally tracked my sleep for many years with my phone - it's easy to do, and provides me with valuable insight. However, there is no way currently for me to add this information to my medical record - despite being information about my health.

Statistical Analysis

Patient-facing healthcare software has the potential to change how we consume healthcare - or maybe it won't make a difference at all. But either way, without having the systems in place to attempt it, such as OAuth logins, we won't find out.

There is a wealth of potential discoveries to be made through statistical analysis of medical records. In 2012, a team of three discovered unique drug interactions by analysing open medical records provided by Stanford University (Tatonetti, Ye, Daneshjou, & Altman, 2012). Despite using only statistical methods, they found suggestion of a biological interaction.

Ensuring that it's easy to collect the information for this type of analysis to undertaken is worthwhile. The value of this is not only for researchers, but public health experts, institutions, and policymakers. Being able to easily collect large-scale medical information which can be easily analysed or represented has value for many stakeholders.

Automatic anonymisation tools would be a simple way to facilitate this. This tool would automatically remove personal, or identifying information from records. It would allow institutions to easily hand over, or publish data.

User Interface

User Interfaces (UI) facilitate the interaction between the user, and the software. They are a metaphor, providing a representation of the program designed for human use. A bad UI can make fantastic software useless, and a good UI can make a program an indispensable tool.

So when designing an interface, a question worth considering is,

What makes an interface good?

In his book "The best Interface is No Interface", designer Golden Krishna (2015) attempts to tackle this question head on. His point is that the user interface is a point of friction between the user and their goals. Users don't actually want to click, tap, or chat with an interface. They want to check their bank balance, communicate with friends, or find out if their patient is allergic to penicillin.

He also points out downsides of modern interfaces - they can be distracting, and addictive. The average smartphone user checks their phone 150 times a day. Doctors are also very conscious of distractions. Bryan for example, who I interviewed has his setup specifically organised to stay focused and on task.

Health is also a concern. Digital screens are essentially a giant lightbulb, and using a screen 8 hours a day is a well-documented cause of insomnia.

Krishna suggests a "NoUI" future, and discussed three principles to help get there.

1. Embrace typical processes instead of screens

By 'typical processes' he means the typical processes of the user. By paying acute attention to how the user functions, you can build an experience which works with the user. He gives the example of a product called 'Square' (Square Inc., 2011). Square provides a Point of Sale terminal for the shop assistant to complete transactions, allowing the user to pay via credit card - or their smartphone.

They have a feature called 'auto tab'. You as a user walk into your regular coffee shop. Your phone sends out a Bluetooth signal, informing the Barista that you are in the shop, and that the last ten visits you ordered a cappuccino. After you pick up your desired item, payment is automatically debited and your receipt is emailed to you.

Obviously you'd only use this feature at shops you regularly visit, but consider the change in experience.

When a task has to be repeated commonly, any way to reduce the steps involved is worthwhile. In this example, Square took it as far as possible, and completely removed the user's interaction with the interface.

2. Leverage Computers Instead of Serving Them.

Computers are incredible, able to process information significantly faster than humans ever could. However, as much as computers serve us - we also spent a lot of time serving them. If an interface is how software communicates to us, then we are the interface between the software and the world. We are essentially their senses - software knows nothing about the reality we live in, other than what we've told them. He argues that software should use digital senses as best as possible, to reduce the need for user input.

Software also creates "digital chores", tasks which the software requires us to do. Passwords to reset, notifications to attend to, files to sort, messages to archive, calendars to update - the digital maintenance of our lives. His argument is that good systems should remove chores, not create them.

3. Adapt to Individuals.

Software is traditionally made for the average user - however, no user is average. Software however can adapt to individuals by learning about them. This provides incremental gains, but the more it learns the better it can serve you.

He gives the example of Nest (Nest Labs, 2011), a digital thermostat. It functions like any regular thermostat, you turn the temperature up and down as you please. However, it learns and notices common patterns - perhaps you always turn it up when you wake up at 7 am, perhaps you prefer a temperature of around 19 degrees. Eventually, the thermostat will manage itself.

Interfaces for EHR software.

I think interfaces for EHR systems need to have one key quality - adaptability, for three reasons:

User Context

As I explored in Storage, patient's have varying records. A newborn's record looks very different to an elderly cancer patient. And beyond that, what information in that record is relevant will depend on which medical professional is treating them. A doctor in an emergency room will be looking for very different information than an optometrist.

Karsh et al (2010), concluded that a user's needs of their software can change tremendously between clinical role, situation, environment, and institution. When I interviewed Bryan (A GP), he noted that while sharing a single record with other institutions would be useful, a lot of the information would be irrelevant to him - even if it's very relevant to another person treating the same patient.

Ultimately, the interface needs to be designed with context in mind - and therefore, the UI needs to be able to adapt to many different situations.

Optimisation

Designing a user interface to say, prescribe medications, may seem simple. But it's worth considering the impact - A GP could potentially have to prescribe medications 160 times a week. At that scale, it's worth considering efficiency, and safety. This requires constant redesigning, tweaking, and user feedback.

Optimisation is a long process, and requires many iterations, and adaptations of the interface. Having an interface designed for adaption would make the iteration loop of designing, building and then testing - much faster.

Individualism

As demonstrated by Bryan's interview, GP's will go to great lengths to optimise and customise how they work. Bryan has his setup which he has designed for himself - but not every GP would want to work like that. They're all unique users, with unique preference as to how they should work.

If I as a designer, design a UI which cannot be changed at all, I am forcing my opinions of how to achieve a task onto my users. Ultimately, the design choices I make will affect how the end user will act, as noted by Karsh et al (2010).

"Users are inevitably and often unknowingly influenced by what many HIT designers might consider trivial design details—placement (information availability), font size (salience), information similarity and representativeness, perceived credibility (or authority)"

So, if I make design choices which are unchangeable by the end user, I am essentially forcing my opinion onto medical professionals - despite, not being one myself. Even if the interface is designed alongside a group of GPs, there are still edge cases. Some medical professionals have poor eyesight, some prefer using the mouse over keyboards, some couldn't get any work done without their keyboard shortcuts.

Users are individuals, and the UI must adapt to their unique workflows.

Precedents

Adaptable interfaces aren't new - here are some examples.

Email

Email is the worldwide standard for sending electronic mail. As users of email, it's accessible everywhere - on our computers, on our phones, and on our watches. While the functionality of email remains the same between platforms, email clients exist for almost any device connected to the internet.

Email was designed with a very clear separation between the server, and the client. The email server provides the functionality of storing and sending email, and the client provides an interface for the user.

This separation works well because anybody can create a new email client, and get users to try it out, without changing anything about the email server. Dislike the layout, want a new feature, want a workflow adapted for your smartphone? Go for it! This model of open opportunity to redesign the end user interface, means end users have a huge range of options.

Atom Text Editor

Atom describes itself as "A hackable text editor for the 21st century". Atom is simply a text editor, designed for programmers. However, what makes it unique is how much users can change about it. Users can write 'packages' which change part of Atom - either simple changes to the interface appearance (such as increasing the font-size), to adding new functionality (such as automatic code checking). Users have customised Atom to provide almost any feature you could imagine - from controlling your Spotify music inside Atom, to specific workflow improvements such as Hey Pane. As of writing, there are 6,244 packages published, which any user can download and add to their editor.

This 'extensible' approach provides some benefits over the adaption of email clients. Firstly, if you just want to make a small change you don't have to start from scratch. In order to make a change to an email client, you have to build an entire email client from the start. Atom provides a simple way to make small changes, without affecting the majority of the software. Secondly, it lets users customise as much as they want. As not all users work the same, Atom's approach of mixing and matching packages means that any user can create a setup which works for them.

Similar approaches have been successful for other pieces of software, such as Chrome Browser Extensions, Sublime Text, or Sketch Plugins.

Design Languages

If interfaces can be adapted by their users down to the smallest detail, they run the risk of becoming inconsistent. If an adaption works in an unexpected way, they become confusing and hard to use - and by letting users customise them down to the smallest detail, you run this risk.

That's the issue design languages solve. A successful precedent is Apple's Human Interface Guidelines. This document describes a consistent look and feel, which all iOS apps must adhere to. It not only has aesthetic requirements, but design principles, and standard interactions. This creates consistency - a date picker will look and act the same in one app, to every other app. This helps make apps easy to use, any user can download and use a new app with limited or no instruction. This is because while the app is new to the user, the UI elements, and interactions, are not.

Summary

In order for an interface to cater to the large range of user needs in Healthcare, it must be adaptable. The best way to do this, is with an extensible interface, providing mix-and-match customisation. However, in order to maintain consistency, and therefore ease of use - a design language is required. This language is a set of design principles, interactions, UI elements, and aesthetic choices.

Output

My output is a design proposal for a fictional EHR system, called Barnett. My proposal is shown on the homepage of this website. It is a combination of what I think are the best solutions to each of my problem areas.

Format. The file format is a modular, extensible format, which is both consistent but easily adaptable to user needs.

Authentication & Privacy. Barnett uses a flexible authentication system to adapt to whatever is appropriate. 3rd party apps can access information through OAuth logins, and statistical tools are provided for research.

Network. A distributed network simulates a single record per patient - while providing a robust, privacy conscious implementation.

User Interface. An extensible interface provides a framework which can be adapted to the context of care, and individual user.

I think Barnett has two unique design philosophies behind it.

Problem Focused.

Barnett isn't focused on the needs of any specific organisation, instead, it is focused on the needs of the problem - the management of health records. It provides an agnostic way to store, share, and interact with health records. This means it has limited scope of functionality compared to EMR systems used in Primary Care - for example, it doesn't provide any appointment management or invoicing tools. It's designed to be used in unison with other tools which will fill this gap.

I believe that by splitting up the functionality provided by EMR systems, there will be more opportunity for product designers and developers to improve the experience of health care professionals, and specialise on hard problems.

Adaptability

At its core, Barnett accepts that healthcare is varied. The needs of users change dramatically depending on context, institution, region, and the individual. Instead of working against variety, Barnett is designed to adapt to variety. The format of the records themselves can be easily adapted, and shared. Likewise, the interface is extensible, and customizable down to a very specific detail.

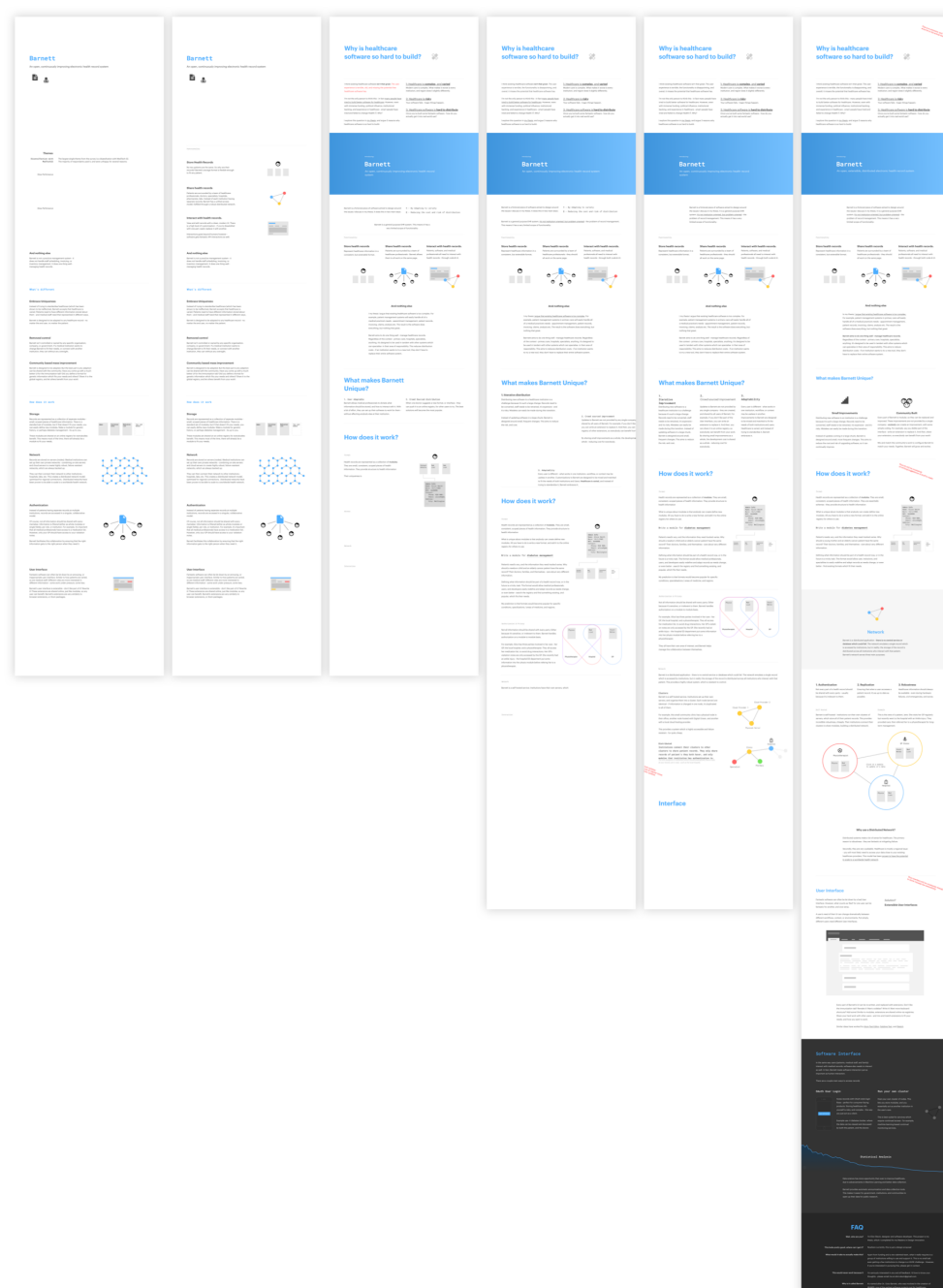
Extensibility is a long celebrated factor of good software. In fact, The Internet Engineering Task Force (the group of individuals who oversee the creation of the protocols and technologies that power what we know as the Internet), note that extensibility is a key factor in a protocol being 'Wildly Successful' (Aboba & Thaler, 2008).

Due to the varying nature of healthcare, I believe that adaptability was a necessary design philosophy for Barnett.

Proposal

Designing the system is one challenge, but explaining it to a visitor, who may not have much experience in Healthcare, or Software Design, is another challenge.

To solve this I used an iterative method of testing with friends and fellow students, and then redesigning. This is appropriate as I wanted the design proposal to be approachable by anybody, not just people with a background in healthcare. The design went through five iterations before settling on the current output.



Critical Review

My output fulfils what I was looking to design as set out in my brief. However, it is not without downsides.

As discussed in distribution, Health Institutions have to weight the cost + risk of upgrading their software against the potential value provided by the new system - there is a socio-technical impact. Barnett's approach to this is to reduce the risk, and cost of upgrading software, with it's extensible, iterative design. Changes can be implemented with minimal disruption or risk. However, what it doesn't solve is how to overcome this problem for the first time implementation.

As discussed in "Electronic health records: research into design and implementation", implementing an EHR system is a large sociotechnical change (Beasley, Holden, & Sullivan, 2011). EHR systems are not simply cleanly added onto existing systems, they require a full transformation of not only the software, by workflows and processes used in healthcare institutions. The impact of implementing such system is not only technical in nature, but societal.

While Barnett provides a clean way to iterate upon itself with minimal impact, its does not address how it could be cleanly built upon, or perfectly replace, existing systems. The initial implementation would still be a huge socio-technical change for an institution. Researching how an institution could make this transition easier was simply out of scope for this thesis. It would require a close, long-term working relationship with a large range of institutions.

However, this is an area for further research, for both Barnett and EHR systems in general - "How can institutions make smoother migrations to EHR systems?".

Related is the second issue - the cost. Developing a system like Barnett would take a lot of development time and testing. In order for a primary care institution to switch to Barnett, they also require additional functionality, such as appointment management. Barnett was designed with the intention that separate software and services would provide that functionality, due to the advantages of compartmentalising complexity. However, for an institution to use this system, that functionality must exist, so these services must also be created.

This of course, increases the cost by a lot. The question then is, is it worth it? In distribution, I question the value provided by better software. While there are undoubtedly benefits to EHR systems, when the cost can easily hit several billion dollars, is it worth it? Will it really have the transformative impact on healthcare that you'd expect from a billion dollar budget?

I think long term a system like Barnett would reduce the cost. However, the upfront spending is high, and it's unclear what party should bear the cost - institutions, patients, or the government?

There is also a potential for a critique of methods. Finding medical professionals willing to spend their precious time to interview with me, was a challenge. In the end, I found four out of six of my

candidates from a GP IT research group. This research group was a group of GPs who had self-selected an interest in Health IT.

This could potentially create bias - the individuals I talked were clearly passionate about Health IT, and because of that their opinions cannot be seen as a fully truthful representation of GPs throughout New Zealand. They might have been much more dissatisfied with their tools than other GPs, and had much grander expectations of what their software should do.

The future of Health IT

After completing this thesis, I believe there are three potential futures for Health IT.

1 - Nothing changes. Software is continued to be made for healthcare in exactly the same fashion - with disappointing outcomes, slow production speed, and high cost.

2 - Healthcare software is made with better processes. The industry as a whole starts improving the methodologies around building healthcare software, and thinks long term. For example, complex systems are broken into smaller systems, vendor lock-in is reduced, and faster iteration loops are set up.

3 - A leap frog technology. A technology comes along which provides either incredible value, or can be implemented with little cost or risk. Modern Machine Learning and Artificial Intelligence have the potential to be this technology, but there are huge challenges to this becoming a reality.

Barnett is firmly centred in the second future - a future where we're not leveraging huge budgets, or futuristic technology, but collectively building with a common goal and methodology of iterative improvement. What future will succeed? Only time can tell.

By Eliot Slevin

Bibliography

Aboba, B., & Thaler, D. (2008). What Makes For a Successful Protocol? Retrieved October 12, 2016, from <https://tools.ietf.org/html/rfc5218>

Amatayakul, M. (2009, March). EHR versus EMR: what's in a name? *Healthcare Financial Management*, 63(3), 24.

Apple Inc. (2014). HealthKit - Apple Developer. Retrieved July 6, 2017, from <https://developer.apple.com/healthkit/>

Ariel Ekblaw, A. L., Asaf Azaria, Thiago Vieira. (2016). MedRec: Medical Data Management on the Blockchain (Version 57e013615dbf3f3300152554).

Auger, J. (2013). Speculative design: crafting the speculation. *Digital Creativity*, 24(1), 11–35. <https://doi.org/10.1080/14626268.2013.767276>

Beasley, J. W., Holden, R. J., & Sullivan, F. (2011). Electronic health records: research into design and implementation. *The British Journal of General Practice*, 61(591), 604–605. <https://doi.org/10.3399/bjgp11X601244>

Brandom, R. (2017, May 12). UK hospitals hit with massive ransomware attack. Retrieved July 3, 2017, from <https://www.theverge.com/2017/5/12/15630354/nhs-hospitals-ransomware-hack-wannacry-bitcoin>

Coiera, E. (2004). Four rules for the reinvention of health care. *BMJ : British Medical Journal*, 328(7449), 1197. <https://doi.org/10.1136/bmj.328.7449.1197>

Compass Health. (2016). Shared Care Record. Retrieved July 3, 2017, from <http://sharedcarerecord.org.nz/>

Department of Health. (2011). Dismantling the NHS National Programme for IT. Retrieved March 27, 2017, from <https://www.gov.uk/government/news/dismantling-the-nhs-national-programme-for-it>

Design Council. (2015). The Design Process: What is the Double Diamond? Retrieved July 3, 2017, from <http://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>

Galloway, A. (2004). Protocol.

Guest, G., MacQueen, K. M., & Namey, E. E. (2011). *Applied Thematic Analysis*. SAGE.

Hardt, D. (2012). The OAuth 2.0 Authorization Framework. Retrieved July 6, 2017, from <https://tools.ietf.org/html/rfc6749>

HealthOne. (2016). HealthOne – Shared knowledge – better, safer care. Retrieved July 3, 2017, from <https://healthone.org.nz/>

HL7. (2016). Index - FHIR v1.0.2. Retrieved December 19, 2016, from <https://www.hl7.org/fhir/>

Indici. (n.d.). Indici | Patient Management System | Designed by GPs for GPs. Retrieved July 3, 2017, from <https://www.indici.co.nz/>

Intrahealth. (n.d.). Profile. Retrieved July 3, 2017, from <http://intrahealth.com.au/our-products/profile/>

Kakouros, N. (2013). Distributed Storage Healthcare — The Basis of a Planet-Wide Public Health Care Network. The Open Medical Informatics Journal, 7, 1–7. <https://doi.org/10.2174/1874431101307010001>

Kamarudin, N. H., Yussoff, Y. M., & Hashim, H. (2016). Passwordless Authentication in Mobile e-health using a Secure Boot Non-regenerated Unique Identity and NFC. Indian Journal of Science and Technology, 9(S1). <https://doi.org/10.17485/ijst/2016/v9iS1/106842>

Karsh, B.-T., Weinger, M. B., Abbott, P. A., & Wears, R. L. (2010). Health information technology: fallacies and sober realities. Journal of the American Medical Informatics Association, 17(6), 617–623. <https://doi.org/10.1136/jamia.2010.005637>

Krishna, G. (2015). The Best Interface is No Interface.

Martin, B., & Hanington, B. M. (2012). Universal methods of design : 100 ways to research complex problems, develop innovative ideas, and design effective solutions. Beverly, MA: Rockport Publishers.

McCluskey, P. (n.d.). Partners launches \$1.2 billion electronic health records system - The Boston Globe. Retrieved March 27, 2017, from <https://www.bostonglobe.com/business/2015/05/31/partners-launches-billion-electronic-health-records-system/oo4nJJW2rQyfWUWQlvydkK/story.html>

McIlroy, D. (1978). The Bell System Technical Journal.

MedTech. (n.d.-a). Medtech Evolution General Practice - New Zealand. Retrieved July 3, 2017, from <http://www.medtechglobal.com/global/products-global/evolution-gp-global/>

MedTech. (n.d.-b). Medtech32 - New Zealand. Retrieved July 3, 2017, from <http://www.medtechglobal.com/global/products-global/medtech32-global/>

Ministry of Health. (2017). National Health Index. Retrieved July 4, 2017, from <http://www.health.govt.nz/our-work/health-identity/national-health-index>

myPractice. (n.d.). myPractice – Smart Electronic Medical Records. Retrieved July 3, 2017, from <http://mypractice.co.nz/>

National Audit Office. (2006). Department of Health: The National Programme for IT in the NHS. Retrieved March 27, 2017, from <https://www.nao.org.uk/report/department-of-health-the-national-programme-for-it-in-the-nhs/>

Nest Labs. (2011). Meet the Nest Learning Thermostat. Retrieved July 6, 2017, from <https://www.nest.com/thermostat/meet-nest-thermostat/>

npm, Inc. (2014). Node Package Manager. Retrieved July 3, 2017, from <https://www.npmjs.com/>

Pandya, D., Narayan, K. R., Thakkar, S., Madhekar, T., & Thakare, B. S. (2015). An Overview of Various Authentication Methods and Protocols. International Journal of Computer Applications; New York, 131(9). <https://doi.org/http://dx.doi.org/10.5120/ijca2015907389>

Practice Fusion, Inc. (2014, May 9). Practice Fusion. Retrieved April 14, 2016, from <http://www.practicefusion.com/>

Randall, M. (n.d.). xkcd: Standards. Retrieved July 8, 2017, from <https://xkcd.com/927/>

Robert, W. (2015). The Digital Doctor.

Rudin, R. S., Bates, D. W., & MacRae, C. (2016). Accelerating Innovation in Health IT. New England Journal of Medicine, 375(9), 815–817. <https://doi.org/10.1056/NEJMp1606884>

Snyder, B. (2013, May 2). How Kaiser bet \$4 billion on electronic health records -- and won. Retrieved March 27, 2017, from <http://www.infoworld.com/article/2614353/ehr/how-kaiser-bet--4-billion-on-electronic-health-records----and-won.html>

Square Inc. (2011). Square - Accept Credit Cards Anywhere. Retrieved July 6, 2017, from <https://squareup.com>

Storj Labs Inc. (2014). Storj - Decentralized Cloud Storage. Retrieved July 4, 2017, from <https://storj.io>

stuff.co.nz. (2015). Elderly man dies after doctor wrongly prescribes Voltaren. Retrieved April 3, 2017, from <http://www.stuff.co.nz/national/health/69799754/elderly-man-dies-after-doctor-wrongly-prescribes-voltaren>

Tatonetti, N. P., Ye, P. P., Daneshjou, R., & Altman, R. B. (2012). Data-Driven Prediction of Drug Effects and Interactions. Science Translational Medicine, 4(125), 125ra31. <https://doi.org/10.1126/scitranslmed.3003377>

The University of Auckland. (n.d.). About thematic analysis. Retrieved July 3, 2017, from <https://www.psych.auckland.ac.nz/en/about/our-research/research-groups/thematic-analysis/about-thematic-analysis.html>

Torvalds, L. (2005). Git. Retrieved July 3, 2017, from <https://git-scm.com/>

Whānau Tahi. (2014). Whānau Tahi | Empowering Connected Care. Retrieved July 3, 2017, from <http://www.whanautahi.com/>

Winton, R. (2016, February 18). Hollywood hospital pays \$17,000 in bitcoin to hackers; FBI investigating. Los Angeles Times. Retrieved from <http://www.latimes.com/business/technology/la-me-ln-hollywood-hospital-bitcoin-20160217-story.html>

ZDoggMD. (2015). EHR State of Mind | #LetDoctorsBeDoctors | ZDoggMD.com. Retrieved from https://www.youtube.com/watch?v=xB_tSFJsjsw

Appendix

This is a table of all 19 responses to my survey. I've removed any personal information that could identify the respondents.

Thanks again to the GPs who took the time to fill this out.

Profile for Mac	As GP practice is so complex with its interactions with so many organisations the PMS has to be able to interact with DHBs, Private organisations, Insurance companies, ACC, PHO, MOH (Immunisations, GMS), NHI look up, eSA, eSAM, GP2GP transfers etc	Intuitive & user friendly - its a MAC!	Have the ability to add "auto complete" PDFs by the practice its self	I hope you have seen P4M in a real live setting it would take something really special to beat it! Oh AND YES I am the alpha tester for P4M & the primary β tester too
MedTech 32	It doesn't integrate properly with all the resources/referral stuff we have, every piece of data has to be manually entered rather than extracting from documents itself, there is scope for a lot more automation with incorporation of clinical guidelines into a PMR to ensure patients' results are appropriately actioned and that treatment plans meet current gold standards (I believe our PMRs should even out variability in practice between GPs so that every patient can achieve the same clinical outcomes). Aesthetically MedTech as a system looks old and outdated - even MedTech Evolution doesn't look like a modern system.	Relatively intuitive - not complicated to use.	See above - modern appearance, better integration, more electronic forms rather than having to print out paper ones, integrated best practice guidelines to assist with management.	
Medtech	Bloody Norah. There isn't enough space.	It's better than paper...	Probably best to interview me. [Note: as the survey was anonymous, I was unable to interview this person, as I don't know who they are]	Oh so many thoughts.
Medtech 32	When we changed hardware Medtech threw up a lot of error messages; it took a long time with escalation of enquiries at the support desk to realise that there was a very quick fix. The error messages and numbers were of no use to identify this problem	good storage of patient records and connectivity with external agencies and the patient themselves (patient portal)	Make it more modern rather than working on a 1990 frame with things being added on	

Medtech 32 2 yrs and profile for mac 24 yrs	I basically hate Medtech their support is appalling the attitudes of users is poor I spent 4 yrs as a beta tester on profile helping to progress the new version one can't do two things at once on Medtech . The inbox is extremely inefficient and time wasting for GPS it's logic is diabolical	I much prefer profile and our practice will be changing from Medtech in the near future. Everybody makes work around son Med tech to get functionality that should be inbuilt but isn't!	Out of Medtech . However Evolution appears equally poorly designed and full of problems	The hospital pms supplier Orion is a web based programme but uses internet explorer 10 which only works with pcs not androids or macs or Linux . As well Ms is not going to support IE and by definition it isn't based on standard web browser protocols
Medtech	Software lacks ability to add in evidence-based topics to guide diagnosis and treatment decisions. Also external patient information such as hospital lab/pathology/outpatient information is not available within the EHR. Also there is no easy way to assess patient adherence to medication (ie graph/gantt diagram of medication given and when it should run out).	Most other parts are OK	Addition of learning or evidence topics in the workflow of the EHR and gantt chart of medication use	nil
Medtech	Not integrated with other health care services...eg secondary. Lots of add ons which slows software	Relatively stable	Integrate the current modules....likely to consider changing to midlands new software and suggest you review this before continuing	Go for it. My email is [removed]
MT32	I thought you said this would be a quick survey! MT32 is very difficult to extend, it both constrains user input terribly whilst allowing so much variation in recording of data that information is hard to transfer between records.	Well, it does what it says on the tin - it is a competent EMR.	MT32 is obsolete, as are most constrained systems. Extensibility at no extra cost is a vital aspect of all PMS these days.	Contact me. [removed]

Medtech	<p>Today for instance, I had the plug-in web based applications non functioning. This results in a few minutes of wasted time per time of use. Sometimes more than one incident per consultation. These were ACC forms and lab request forms. Yesterdays problem was the prescribing assistance for Pradaxa hanging for 5 minutes. Many of my colleagues are not prepared to wait this long, crashing out of the patient medical record system and writing prescriptions by hand. Hence the drug with high risk of adverse effect if not properly monitored does not appear on the electronic medical record at all. Which is worse than useless, and creates medical risk for the patient and legal risk for the doctor and their hapless colleague. (ask me about #PTI files being directed to the wrong person despite numerous requests, again resulting in long computer "hangs" 'till the cloud based applet decides it's going to work.</p>	<p>Overall a structured, legible computerised medical record allows for better collaboration between colleagues and better analysis of trends such as frequency of recurrences, "have they had this before?" medicines utilisation and more.</p>	<p>Have a field to show when patient demographics were last updated and a prompt to check if not up to date. Similar prompts would be helpful for long-term medications and "classifications" (problem lists)</p>	<p>Some of our communications to the secondary sector still require paper to be printed and presented to a certain person in a physical in-tray, where they may be lost, misdirected or ignored with no audit trail. This seems almost wilful from our end, as if barriers to communication were a good thing. Why do we still need to use a (non auditable) fax machine in this day and age?</p>
---------	--	--	---	---

MedT ech3 2	Inflexible. Not designed by clinicians. Not particularly intuitive. (necessary) linkages with other agencies clunky and unwieldy. Poor clinical coding structure - using a cut-down, outdated system taken from the UK. It doesn't utilise the capabilities of modern IT.	It works most of the time. Easy to use basic functionality. However, there is no advance functionality beyond that. The system has not changed significantly since introduction 15 to 20 years ago	Almost everything.	Your model for a record management system does just that and nothing else. We need one that does the other things such as billing, integration with the MOH, PHO, DHB, WINZ, BPAC, NZF and ACC for starters. ePrescribing is hopefully just around the corner and that will DEMAND interaction with some of these agencies so that eScripts can be exchanged between prescribers and dispensers. I'm looking for an end-to-end solution where these integrations have been enabled.
medt ech	There are several issues. The most significant is the ability to accidentally write in the wrong patients notes and the ability to change prescription items accidentally	Using a computerised system has revolutionised medical record keeping and keeping track of a myriad of information about a single patient - and finding information rapidly	It needs to work for me and the patient. So it needs to be safe - and enable me to keep track of important actions easily and unobtrusively	
Medt ech	It is in a cumbersome windows 95 format ,it is difficult to easily see previous consultations and it is prone to frequent error messages, the document system is very limited. There were drug transcription errors in GP2GP notes transfers which could have had serious consequences for the patient and Medtech were very slow to sort out	having electronic results and being able to do e referrals, although those are bolt ons really	speed, format of consultations	It is a dinosaur system and the developers have been sitting on their backsides because of a lack of decent comptitions

Medtech	It works well and I am expert in medical informatics, so quite proficient in the use of electronic record systems. I also do the training and some of the troubleshooting for our organisation in using Medtech. In the U.S. I was a super user of Epicare and helped develop a format for its use in community health centres. Medtech works pretty well.	1. Excellent query building tool. 2. Decent interface with all the different MOH funded programmes. 3. Affordable	I would limit the pathway for external programmes to interface with Medtech so that it would streamline the access point for clinicians. Make it all one sign on and password, not several. Make it one tab/drop down not multiple entry points.	I think that you are starting way behind the leading products and will be facing tough competition.
profile for windows	profile doesn't intergrate with any of the bpac products which are written for medtec 32	very flexible, windows based so very intuitive, can have multiple windows open at once.	have on intergation engine so I can connect to products that run on medtec 32	much better product than medtec 32 but because medtec 32 is the dominant system in NZ , so called national systems are all written for medtac products without thought to other systems
Medtech and Indici	Trying to audit my practice and writing simple query builders is so frustrating - I just dont trust the data I can produce on Medtech.	Indici gives me a holistic view of the patient on a single screen which is really useful to avoid the computer interfering with the consultation	Medtech - I would change it so that it was less Dos and more Windows	nope
Medtech	Frequently occurs. Manage my health making appointments for patients in patient portal but not actually booking the slot in medtech so patients were turning up for appointments that didn't exist	The audit trail when notes are altered	Make the appointments template much more flexible to being altered and a rostering module	Flexibility is key. Practices all work differently and want to tailor their PMS to how they work
Medtech	Often...when it runs too slowly or its association with other systems (eg Best Practise, healthpac) make the system run even slower / don't work at all.	Records are available with just a couple of clicks of the mouse.	Improved interface with other systems that we have to use-ACC, Healthpac, WINZ	

Medtech	<p>When I've written then edited a long letter then lose it. It crashes and closes when i first start it up each morning. Due to an administrator overview thing medtech have added. The other day finished a long consult entry in daily record - and due to an error it all disappeared so i had to start again. The laboratory have changed all the test names, so they no longer match those in the past so medtech wont list them together . I could go on and on.</p>	<p>I know it, and used it for years, it is all simple. Other than hiccups on a single PC the system is stable. Locums and all new doctors and nurses have used it before. But I wouldn't say there is anything really useful about it.</p>	<p>Be able to write on a tablet with a pen, while talking to a patient and for the text to appear into the notes. May be add a small drawing if needed.</p>	<p>Any new software has to be reliable, and secure. IT stuff just seems to have too many bugs. Partly because too many agencies are involved. Govt, ACC, PHO, DHB, WINZ, etc -Need something more like Apple, where others don't get to change bits in the software, and it can all be simpler.</p>
Medtech 32	<p>1. I work in a large practice (30 gp's) and it takes a long time no less than 6-8 seconds per item to prescribe a medicine when I press the F10 button 2. Web enabled forms are great but in practice slow down our workflow. A system that enables a local cache may help them to load faster 3. I would love to be able to tick a combination of typically used medicines to facilitate easier / faster prescribing. 4. Manage my health as a patient portal has failed because it has forgotten who drives utilisation: patients. It needs to be easier to register than the current convoluted process. 5. we have never run a medtech update without disrupting our system and requiring a patch</p>	<p>1. Stable Product. 2. facilitates access to hospital workstation that has revolutionised patient consultation 3. Integrates well with all third party solutions in NZ very well like BPAC, NIR etc</p>	<p>Ability to add combo's of frequently used medicines A faster database or one that allows a larger paging file for larger practices to speed up the application</p>	<p>Medtech is slow and old but it is familiar and very reliable.</p>

MEMORANDUM

TO	Eliot Slevin
COPY TO	Walter Langelaar
FROM	AProf Susan Corbett, Convener, Human Ethics Committee
DATE	22 December 2016
PAGES	1
SUBJECT	Ethics Approval: 23478 Electronic Health Record Software Design

Thank you for your application for ethical approval, which has now been considered by the Standing Committee of the Human Ethics Committee.

Your application has been approved from the above date and this approval continues until 3 August 2017. If your data collection is not completed by this date you should apply to the Human Ethics Committee for an extension to this approval.

Best wishes with the research.

Kind regards



Susan Corbett
Convener, Victoria University Human Ethics Committee