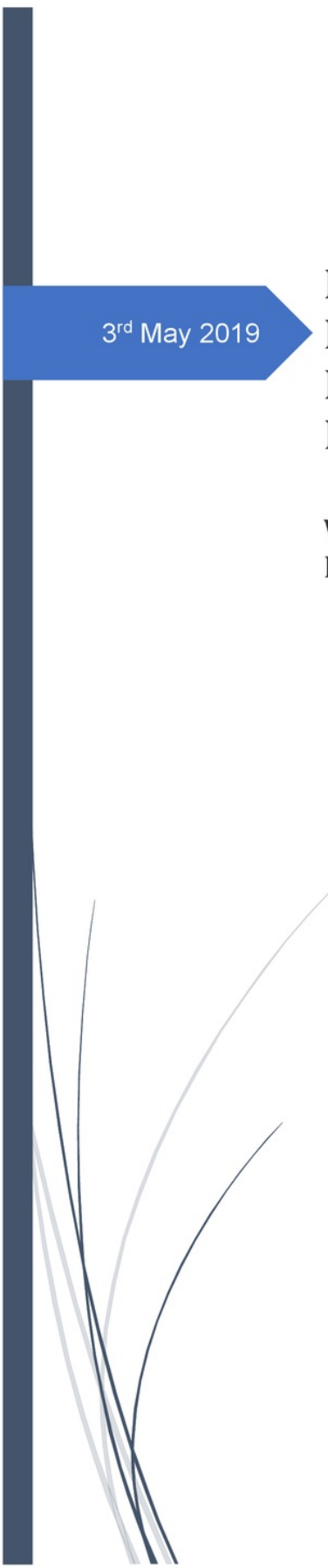


Research Proposal

by Matthew Lewington

FILE	4069_MATTHEW_LEWINGTON_RESEARCH_PROPOSAL_50131_6623062 15.DOCX (458.2K)		
TIME SUBMITTED	27-APR-2019 07:07 PM (UTC+0100)	WORD COUNT	4807
SUBMISSION ID	105506790	CHARACTER COUNT	29056



3rd May 2019

Developing a Smart Automated Medication Dispenser to Assist Patients with Cognitive and/or Motor Impairment

Word Count: 3982

Project Supervisor: Dr Nik Whitehead

MATTHEW LEWINGTON

BSc Software Engineering
1703645@student.uwtsd.ac.uk
UWTSD School of Applied Computing

Table of Contents

1. Introduction	1
1.1 Background Literature	1
1.1.1 Alzheimer's Disease.....	2
1.1.2 Vascular Dementia.....	2
1.1.3 Lewy Body Dementia	2
1.1.4 Frontotemporal Dementia	3
1.1.5 Parkinson's Disease.....	3
1.1.6 Prevalence of Dementia.....	3
1.1.7 Medication Adherence	4
1.1.8 Current Assistive Pillbox Technology	4
1.2 Aims and Objectives	5
1.3 Background Research Conclusion	5
2. Research and Development Methods	6
2.1 Research Methodology.....	6
2.2 Development Methodology	7
2.2.1 Waterfall.....	7
2.2.2 Spiral.....	7
2.2.3 Agile	8
2.2.4 Scrum.....	8
2.2.5 Feature Driven Development	9
2.2.6 Chosen Development Methodology	12
2.3 Risks.....	12
2.4 Ethical/Legal Issues.....	12
2.5 Analysis	13
2.6 Deliverables	13
2.7 Research and Development Methodology Conclusion	13

3. Resources and Planning	13
3.1 Hardware Resources	13
3.2 Software Resources	14
3.3 Literature Resources	14
3.4 Gantt Chart/Timeline.....	14
3.5 Resources and Planning Conclusion	16
4. Conclusion	16
5. References.....	17

Table of Figures

Figure 1-1 – Projected increase in number of people with dementia.....	4
Figure 2-1 – Boehms spiral model of the software process	8
Figure 3-1 – Gantt Chart Tasks.....	15
Figure 3-2 – Gantt Chart Timeline.....	16

Table of Tables

Table 1 - Number and percentage of people with dementia by subtype	1
--	---

1. Introduction

This project will be to design and develop a programmable medication box with the aim of assisting patients that are suffering with cognitive and/or movement impairments such as dementia, or any other condition that can impact a person's ability to remember when to take medication or what medication(s) to take. In the case of motor impairments, a requirement would be to automatically dispense the needed medication. We will be looking to build a prototype and test its functionality and suitability in-house, rather than in industry in the first instance.

This project will be titled "Developing a smart automated medication dispenser to assist patients with cognitive and/or motor impairments".

1.1 Background Literature

Dementia is an umbrella term for a number of neurodegenerative diseases (a progressive disease that affects the brain) that can develop as a direct result of an anomalous reaction from the combination of two or more proteins [1].

Some of the more prevalent types of neurodegenerative diseases that can cause dementia are Alzheimer's Disease, Vascular Dementia, Lewy Body Dementia, Frontotemporal Dementia and Parkinson's Disease. As stated by Scheltens et al [2], McKeel [3, p. 6], Hogervost [4] and as reported by Dementia UK [5, p. 53], the most common and most studied neurodegenerative disease is Alzheimer's disease [Table 1].

Table 1 - Number and percentage of people with dementia by subtype

Subtype	Percentage	Number
AD (Alzheimer's Disease)	62	505,813
VaD (Vascular Dementia)	17	138,691
Mixed (AD and VaD)	10	81,583
LBD (Lewy Body Dementia)	4	32,633
FTD (Frontotemporal Dementia)	2	16,317
Parkinson's	2	16,317
Other	3	16,317
Total		807,671

1.1.1 Alzheimer's Disease

In 1907, Alois Alzheimer first defined the characteristics and pathologic attributes of Alzheimer's Disease (AD) in "Über Eine Eigenartige Erkrankung der Hirnrinde" (on a peculiar disease of the cerebral cortex) [4] and was among the first to connect cognitive dysfunction with structural changes within the brain [3, p. 1]. AD is usually characterised by the following symptoms [3, p. 16], [6]:

- Early memory loss,
- Difficulty performing everyday tasks,
- Forgetting words or using substitutes,
- Disorientation to time and location,
- Poor judgement,
- Problems with abstract thinking,
- Item misplacement,
- Mood/behavioural changes,
- Personality changes,
- Loss of initiative.

1.1.2 Vascular Dementia

Vascular dementia (VaD), being the second most common form of dementia, does not initially present with memory loss - this in fact tends to come later in the disease [4]. As stated by Bowler et al. and Tatemichi et al., cited by Dae Kwon and Heinbockel [7], VaD patients are more likely to have less prominent memory impairment, but a larger case of frontal executive dysfunction [3, p. 95] (a problem with a high order cognitive processes [8] that occur in the frontal lobes of the brain) such as planning and organisation, flexible thinking, self-awareness, etc.. [9].

1.1.3 Lewy Body Dementia

Lewy Body Dementia (LBD), whilst the core clinical features include fluctuating cognitive impairment, there are other symptoms such as recurrent visual hallucinations and Parkinsonism [1], [4]. The visual hallucinations present early on when dementia is relatively mild, but they tend to persist [3, p. 124].

Parkinsonism develops in most individuals that are affected by LBD – signs typically include postural instability and bradykinesia, with tremor being less common.

1.1.4 Frontotemporal Dementia

Frontotemporal Dementia (FTD) is characterised by a progressive degeneration of either the frontal or temporal lobes, or both [3, p. 149], [10]. Patients with FTD may not show signs of memory difficulties in the early stages, but they may present with language difficulties such as aphasia [3, p. 150], an uncontrollable repetition of a word or sentence, or repetition of any vocalisation made by a person within range [1] and personality change [4]. Motor issues may also contribute to FTD such as Parkinsonian syndrome and motor neurone disease [3, p. 150].

1.1.5 Parkinson's Disease

Parkinson's Disease (PD) includes dementia as a symptom in up to 40% of patients. PD dementia (PDD) and LBD share clinical and neuropathological phenotypes and are often indistinguishable [3, p. 127]. Currently the two are distinguished using a one-year rule; PDD is diagnosed only if the onset of dementia occurs after 12 months from the diagnosis of Parkinsonism, otherwise LBD is diagnosed [3, p. 127]. PD also has motor impairment as a symptom as stated above such as postural instability and bradykinesia, with tremors in up to 70% of PD patients [11, p. 33]. Movement issues occur before cognitive symptoms, unlike LBD where they occur after, or at the same time as cognitive symptoms.

According to a report by Parkinson's UK [12, p. 18], using results from the Clinical Practice Research Datalink, the number of patients with Parkinson's Disease is set to double between 2015 – 2065.

1.1.6 Prevalence of Dementia

From research by Prince et al into the findings of a report by Alzheimers Society, the prevalence of people with dementia is expected to grow rapidly [Figure 1-1] over the next few decades. This unprecedented growth can be attributed to increased life expectancy in developed countries [3, p. 1].

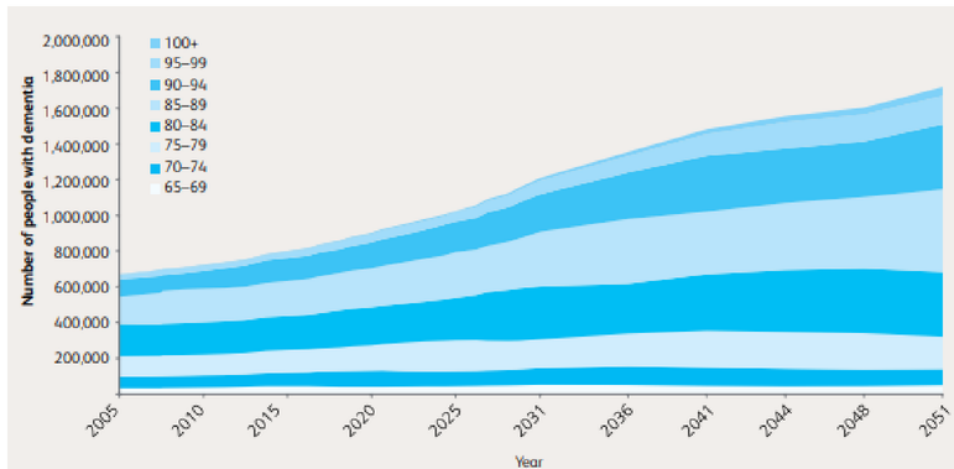


Figure 1-1 – Projected increase in number of people with dementia in the UK by age group [5, p. 46]

1.1.7 Medication Adherence

As cited by Jackson [13], Sabaté states that medication adherence is defined as “the extent to which a patients behaviour in taking medication corresponds with agreed recommendations from a healthcare provider”. Medication adherence is an important challenge for many patients with chronic conditions [14]. In some cases, such as dementia or another condition that can impair cognitive function, it may be the case that the patient merely forgets to take their medication or is in fear that they may have already taken their medication so they would not want to double dose or over dose. A patient with motor impairment may struggle to open the blister packets, or bottle(s) of medication for example.

1.1.8 Current Assistive Pillbox Technology

Compliance aids have been shown to improve medication adherence [15], and there are many forms of this assistive technology available such as the Nomad® MDS [16] which is a medication box with 28 dose blisters, and is available in 4 sizes. These are however quite large and bulky, so may be more difficult to transport and to fill the boxes with the relevant medication.

Venalink® MDS [17], which is smaller than the Nomad® MDS, however it again utilises blister packaging so patients may struggle to retrieve medication from the packs.

More popular designs now are dispensers such as the TabTime® Medelert [18] and Pivotell® Advance [19], both of which are currently expensive solutions although offer much needed functionality.

1.2 Aims and Objectives

The aim of this project is to build a programmable medication dispensing pillbox as previously stated, to assist patients with cognitive/motor impairments with medication schedule adherence. The objectives for this project, in no particular order will be:

1. Investigate current medication schedules for patients with said impairments.
2. Build a prototype hardware for the medication dispensing pillbox.
3. Build a prototype software to allow the medication schedules to be set up and the pillbox to be programmed.
4. Evaluate the pillbox in a controlled, non-consumer environment with extensive testing.

At the end of the project, it is to be expected that a prototype hardware will have been developed, along with accompanying software that is suitable and relevant to the end users' needs.

To accompany the product, a requirement specification sheet will be produced to highlight the primary features and any additional features.

1.3 Background Research Conclusion

From the background research provided it is clear that motor and cognitive impairments can be incapacitating in a number of ways, and that there is a predicted increase in the number of people with diseases that can cause these conditions over the next coming years.

It is important that a medication schedule is adhered to in order to alleviate symptoms and to allow a better quality of life for these patients, therefore it is proposed that development of a smart pillbox will take place to assist these patients in schedule adherence, and also to make it easier for patients with motor impairment to get access to their medication, as they will not need to open any bottles or remove tablets from blister packs.

2. Research and Development Methods

This section will discuss the research and development methodologies and state why they were selected. A brief look at different development methodologies will provide an outline to benefits/drawbacks of different ones and the best suited one will be selected.

2.1 Research Methodology

The methodology chosen to follow for this project will be a positivism philosophy and a deductive approach. Existing theories and principles of electronics and software engineering shall be used, and data gathered will be the results of tests that are implemented to test if the product works correctly along with user experience (UX).

One research strategy to be used will be software (and hardware) development, as the smart pillbox will be created along with an accompanying application to communicate with the pillbox. A second strategy will be a document review, to identify current devices or technologies that are similar to this project, or such studies or products that can help people remember when to take their scheduled medication or assist them with doing so. The third research strategy for the project will be a case study (although an experiment could have been used, ethical and legal approval along with a relevant test subjects would be required, and this could be difficult to obtain). Case studies are usually more appropriate to the phenomenological methodology, but the product is being designed for a specific problem and a study will be required to ensure the pillbox is fit for purpose and the requirements are fulfilled such as does it dispense the medication at the correct time, and is the application easy to use for patients, caregivers, etc..? As the application should be able to be used by people with little to no medical experience such as family members just making sure the patient has taken their medication.

To produce this data, the product will need to be observed and tested throughout development to ensure it adheres to the schedule and correctly dispenses medication. Usability/accessibility for the application will need to be a questionnaire based on UX when using the application.

2.2 Development Methodology

The stages that all development projects go through are all alike, regardless of the methodology that is chosen [20, p. 52]. A development methodology provides a framework, or template, to manage, structure and control a development project [20, p. 55]. They are known as software (or system) development life cycle (SDLC) methods [21], [22], and there are many different methodologies that can be followed, some more appropriate than others. Each of the following methods require similar phases that are followed, which are essential for software engineers:

- Requirements
- Analysis
- Design
- Development
- Testing
- Implementation
- Maintenance

2.2.1 Waterfall

The Waterfall Model is the oldest and universally followed SDLC model [22]. It is sequential, as in it has specific steps through the phases of requirements analysis, design, coding, testing and maintenance, and is meant to ensure design flaws are identified before development begins. However this is largely based on beliefs, not empirical evidence [23]. Due to its linear approach to development, it struggles with change and can generate repeat work, as once a phase is completed it is not easy to go back.

2.2.2 Spiral

The spiral model is a development process that combines the iteration processes of prototyping with the systematic approach that the waterfall model uses, and was the first model to provide information on why iteration is important [24]. Unlike the waterfall model, it is possible to perform the SDLC phases multiple times as shown in Figure 2-1, and each time it is performed is a closer prototype to the final product [22], [25, p. 741].

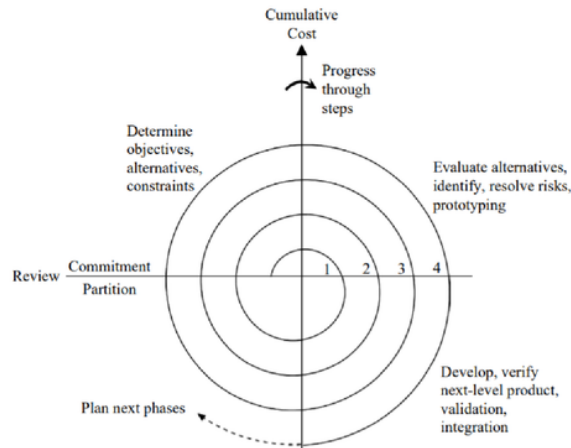


Figure 2-1 - Boehms spiral model of the software process [25, p. 742]

2.2.3 Agile

Agile development can be seen as a reaction to developers' needs when faced to increasingly varying requests from clients, and due to the flexibility of the growing economic environment, organisations need the ability to adapt to these opportunities with ease [26]. The testing phase is integrated with the development phase and they are actively completed together, however a major downfall is that documentation is often ignored, and the core principle is to carry on working on the software product [27]. A key period of agile development is a sprint. Sprints are short periods (2-4 weeks) where teams aim to complete a set amount of work.

2.2.4 Scrum

Scrum is one of several methods of agile software development, which is an incremental methodology. The scrum is a framework where multiple practices and methods techniques are applied. The role of scrum is to highlight effectiveness of software development practices, while also providing an environment in which the development of complex products can be accomplished.

In comparison to the waterfall model, the agile scrum methodology can offer more flexibility and consistency [28]. There are multiple roles in a scrum environment such as the product owner, teams and the scrum master. Again, sprints are utilised to develop an increment of the current software version

which would usually include some form of new functionality or one or more features [29].

2.2.5 Feature Driven Development

Feature Driven Development (FDD) is a model-driven short-iteration process which starts with defining an overall model template, continuing with a number of fortnightly “design by feature, build by feature” iterations [30]. It is client centric which develops software according to client valued features and development emphasises more on quality [31].

Traditionally the FDD life cycle is based on five processes, or stages of development [32, p. 103]. They are:

1. Develop an overall model
2. Build a detailed, prioritised features list
3. Plan by feature
4. Design by feature
5. Build by feature

2.2.5.1 Develop an Overall Model

Within the model development, there are a number of tasks to be completed [32, p. 106]. These tasks are:

- To form the model team,
- Conduct a domain walkthrough,
- Develop small group models,
- Develop a team model,
- Refine the overall object model,
- Write model notes.

Domain and development team members will work together with the supervision of a senior object modeller (chief architect). A primary high-level analysis is conducted of the system scope, then a more detailed analysis is performed on each area.

Small groups are formed, and each respective group then creates a model of the domain area for review from the chief architect. The overall object model is then combined with the main domain area model, and the object model is updated in iterations [32, p. 106].

2.2.5.2 Build a Features List

Once the initial object model is built, the modelling team moves forward to create a list of features. This is completed in two sections. They are to form a feature list team, and then build the feature list. The team tends to consist of just the chief programmers from the previous process, as they will have the knowledge to be able to perform this task correctly. This team then breaks the domain down into smaller areas, and further again into a number of activities. Each step within an activity is a feature [32, p. 135].

2.2.5.3 Plan by Feature

The order of feature implementation is depicted by the Project Manager, Development Manager and the chief programmers. The tasks within this process are flexible, as in they are not a strict sequence as they were previously. They are considered all together, and refinements are made from one task to the other. The main task is to form the planning team. Subsequent tasks, in no particular order are:

- Determine the development sequence
- Assign feature sets to chief programmers
- Assign classes to developers

An initial completion date is outlined by the planning team for each feature. It is based on variables such as feature implementation complexity, bringing forward high-risk or complex feature sets, load-balancing across development teams and dependencies between features [32, p. 146].

2.2.5.4 Design by Feature

There are multiple steps within this phase, where the client-valued function is delivered [32, p. 159]. They are:

- Form a feature team,
- Conduct a domain walkthrough,
- Study the referenced documents,
- Develop the sequence diagrams,
- Refine the object model,
- Write class and feature prologue,
- Design inspection.

The chief programmer chooses which feature(s) to develop from their itinerary of assigned features. A feature team is then formed by identifying the owners of the classes who will be likely to work on the development of the chosen feature(s).

A domain walkthrough is an optional task depending on the complexity of the feature, but if conducted then it will outline data items, algorithms and minor objects involved for the feature team. It can be skipped if there is a small amount of data or the algorithms used are not too complex.

Document study is again an optional task, depending on algorithmic or calculation complexity. The main result from this task is to increase understanding and knowledge for the feature teams of the data items and algorithms required for implementation.

Sequence diagrams are developed by the feature team for each feature being designed. The team members work together to come up with the optimal set of object interactions required to fulfil the requirements of each feature being developed.

To refine the object model, the chief programmer creates a directory on a network, and the feature team can then use this environment to share any current work in progress, and this allows shared visibility amongst the team. The chief programmer then further refines the object model and adds additional classes, operations and attributes or makes changes to any current classes.

Using the updated implementation files from the previous phase, each of the class owners write the classes and relevant function prologues for each item that is defined by the current feature being worked on.

The feature team conducts a design inspection along with other project members.

2.2.5.5 Build by Feature

The final process in the FDD model is known as build by feature. This comprises of the following:

- Code
- Unit testing
- Code inspections
- Promote to build

Activities are completed per feature, referencing the design package that was produced in the previous phase. Class owners implement the required items for the design of the features, unit tests and code inspections are completed (in either order), then the code is promoted to build [32, p. 181].

2.2.6 Chosen Development Methodology

The development methodology selected for this project is the FDD method. The core reason being as the software will primarily be feature built, each iteration of updates will be better than previous, and it tends to emphasise quality which will be key for this project. Works completed, in progress, and required are visible from the very beginning which would provide a clear outline of progress and scheduling should be easier.

2.3 Risks

There are a handful of risks that can pose problems for the project completion. These include

- Failure to code the core software features,
- Failure to understand and implement the 3D printing software,
- Failure to obtain required hardware.

In an attempt to contest with these potential issues, core focus will be on the above as there does need to be a minimum viable product for the project, and as the project is being completed for a purpose it does need to be suitable for the environment in which it could potentially be utilised.

2.4 Ethical/Legal Issues

As the product will not be tested in an environment where it is actually being used by patients, no permissions will be required for this. The only ethical concern would be with the questionnaire, however no personal information will be collected. The results from the questionnaire will be only to gauge usability of the application software. There will be no need to contact the testers afterwards, so no contact information is required.

2.5 Analysis

The data obtained from in-house/development testing would be whether the pillbox correctly dispenses the medication at the required time, and whether the end user finds the application software easy to use.

The pillbox must be able to dispense the medication at the specified time in order to be a success, and the end user must find the application software easy to use as it is meant to be able to be used by persons with no medical expertise. The results from the questionnaire will be used to produce a list of potential future features, or if enough time is available to refine the product itself and release a second version of the product.

2.6 Deliverables

The deliverables of the project will be the pillbox itself, accompanying application software, software and hardware requirement specification sheets, and results from the questionnaire.

2.7 Research and Development Methodology Conclusion

As outlined, the research methodology and development methodologies have been selected to ensure correct research and project development. Risks are manageable as long as the development is persistent and there are no knowledge gaps. There are no ethical/legal permissions required due to the environment the product will be produced and tested in, and any test users will not have their personal information stored. There are multiple deliverables ranging from documents to hardware and to software.

3. Resources and Planning

In order to successfully complete the project, the following resources (broken down into individual sections) will be required.

3.1 Hardware Resources

The main hardware required would be:

- PC/laptop for the development of the system including 3d modelling and coding,

- 3D printer will be needed to build the pillbox,
- IoT kit, along with relevant modules.

3.2 Software Resources

Software required for the project would include:

- 3D modelling/design software to build a prototype pillbox,
- 3D printing software to allow the printer to print the model,
- Software such as Visual Studio, or Arduino Sketch for writing the code.

3.3 Literature Resources

Although some background reading has been completed, further information may be required such as:

- Information regarding medication and schedules,
- Books and journals on the subject area.

3.4 Gantt Chart/Timeline

A Gantt chart has been produced as shown in Figure 3-1 and Figure 3-2, to allow scheduling for project success. It is imperative this timeline is stuck to for the project to succeed. Although it is not always guaranteed that the timeline can be adhered to, such as if problems occur with software/hardware integration or acquiring specific modules for the IoT kit. If one section is completed early, then it is always possible to start a new section early to allow for such hurdles.

Task Name	Duration	Start	Finish
Research	16 days?	Tue 19/02/19	Tue 12/03/19
Background Reading	16 days	Tue 19/02/19	Tue 12/03/19
Literature Review	16 days	Tue 19/02/19	Tue 12/03/19
Product Research	16 days?	Tue 19/02/19	Tue 12/03/19
Existing Products	16 days	Tue 19/02/19	Tue 12/03/19
Requirements & Specification	65 days	Tue 19/02/19	Mon 20/05/19
Identify Aims & Objectives	16 days	Tue 05/03/19	Tue 26/03/19
Produce Methodology	54 days	Tue 19/02/19	Fri 03/05/19
Hardware Requirements	11 days	Mon 06/05/19	Mon 20/05/19
Software Requirements	11 days	Mon 06/05/19	Mon 20/05/19
Web/App Requirements	11 days	Mon 06/05/19	Mon 20/05/19
Design Research	31 days	Mon 20/05/19	Mon 01/07/19
IoT Kit Suitability	31 days	Mon 20/05/19	Mon 01/07/19
IoT Module Suitability	31 days	Mon 20/05/19	Mon 01/07/19
3d Printing Software	31 days	Mon 20/05/19	Mon 01/07/19
Design	11 days	Mon 01/07/19	Mon 15/07/19
Preliminary Product Design	11 days	Mon 01/07/19	Mon 15/07/19
Development	150 days?	Tue 16/07/19	Mon 10/02/20
Hardware Development	150 days	Tue 16/07/19	Mon 10/02/20
Software Development	150 days	Tue 16/07/19	Mon 10/02/20
Web/App Development	150 days	Tue 16/07/19	Mon 10/02/20
Testing	164 days	Tue 16/07/19	Fri 28/02/20
Developer Testing	150 days	Tue 16/07/19	Mon 10/02/20
Code Review	150 days	Tue 16/07/19	Mon 10/02/20
Code Modification	150 days	Tue 16/07/19	Mon 10/02/20
Functionality Testing	150 days	Tue 16/07/19	Mon 10/02/20
User Testing	5 days	Mon 17/02/20	Fri 21/02/20
Questionnaire	5 days	Mon 17/02/20	Fri 21/02/20
Results & Analysis	5 days	Mon 24/02/20	Fri 28/02/20
Questionnaire Results	5 days	Mon 24/02/20	Fri 28/02/20
In-house Test Results	5 days	Mon 24/02/20	Fri 28/02/20
Refinement	11 days	Mon 02/03/20	Mon 16/03/20
Implement Software	11 days	Mon 02/03/20	Mon 16/03/20
Evaluation	24 days	Tue 17/03/20	Fri 17/04/20

Figure 3-1 – Gantt Chart Tasks

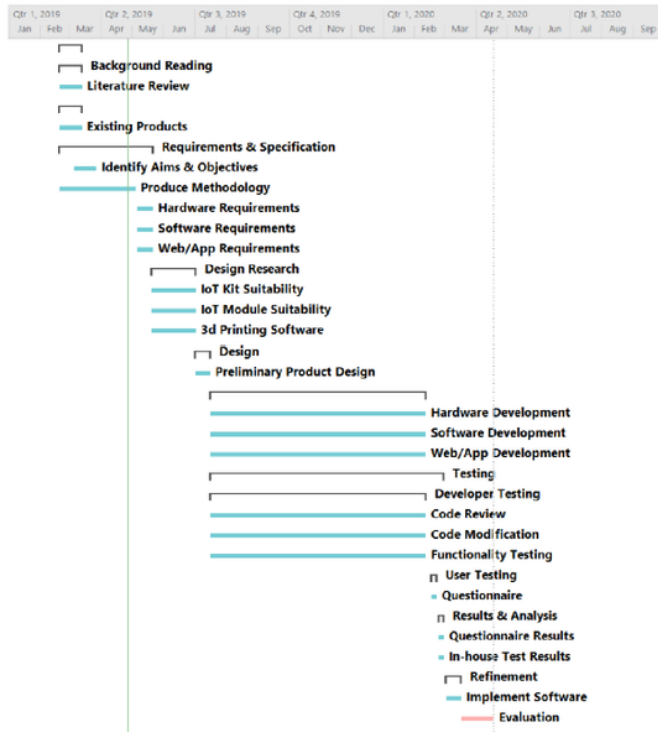


Figure 3-2 – Gantt Chart Timeline

3.5 Resources and Planning Conclusion

There are a number of resources required to ensure the completion of this project. The core resources obviously are the hardware (IoT kit and modules, 3d printer) and the software. These should all be relatively easy to get hold of and should pose no threat to the project. As some time has been spared in the timeline, and previously stated within the research and development section, it would be possible to progress the project if a section has been completed already. Testing will be carried out throughout the project duration to ensure that the product is working upon project completion, as there is no point moving on to develop a new feature when the current feature is not working correctly.

4. Conclusion

As advised, this project will be to design and develop a smart pillbox to assist patients to ensure they are adherent to their medication schedules. This not only improves symptoms, but also can increase the quality of their lives as

symptoms may be more manageable with correct medication dosing and timing, and it will be one less concern in the patient's daily life whether they have taken a medicine or not.

A suitable philosophy and methodology have been decided on to assist with the project planning and completion, and relevant resources have been identified. There should be no problems in completing this project as the risks are relatively low as long as milestones are achieved within reasonable timing.

Overall it is believed that this is a viable major project and completion should be achievable. The timeframe is not unrealistic, nor is the proposed work load too much.

5. References

- [1] C. Holmes and J. Amin, "Dementia," *Medicine (Baltimore)*., vol. 44, no. 11, pp. 687–690, Nov. 2016.
- [2] P. Scheltens *et al.*, "Alzheimer's disease," *Lancet*, vol. 388, no. 10043, pp. 505–517, 2016.
- [3] D. W. McKeel, *Dementia*. Oxford: Oxford : Atlas Medical Publishing Ltd, 2007.
- [4] E. Hogervorst, "Healthy lifestyles to prevent dementia and reduce dementia symptoms," *Work. With Older People*, vol. 21, no. 1, pp. 31–39, 2017.
- [5] M. Prince *et al.*, "Dementia UK: Update Second edition."
- [6] "Symptoms of Alzheimer's disease | Alzheimer's Society." [Online]. Available: <https://www.alzheimers.org.uk/about-dementia/types-dementia/alzheimers-disease-symptoms>. [Accessed: 06-Apr-2019].
- [7] O. D. K. E.-T. Heinbockel, "Cognitive Features of Vascular Dementia," Rijeka: IntechOpen, 2012, p. Ch. 7.
- [8] F. Lange, C. Brückner, A. Knebel, C. Seer, and B. Kopp, "Executive dysfunction in Parkinson's disease: A meta-analysis on the Wisconsin Card Sorting Test literature," *Neurosci. Biobehav. Rev.*, vol. 93, pp. 38–56, 2018.
- [9] "Executive dysfunction | Headway." [Online]. Available: <https://www.headway.org.uk/about-brain-injury/individuals/effects-of-brain-injury/executive-dysfunction/>. [Accessed: 06-Apr-2019].

- [10] J. D. Warren, J. D. Rohrer, and M. N. Rossor, "Frontotemporal dementia," *BMJ Br. Med. J.*, vol. 347, no. 7920, 2013.
- [11] K. R. Chaudhuri, *Fast facts Parkinson's disease*, 3rd ed. Abingdon, Oxford: Abingdon, Oxford : Health Press Ltd., 2011.
- [12] "The prevalence and incidence of Parkinson's in the UK," 2017.
- [13] D. Jackson, "A Myriad of Multiples: Many Problematic Factors Contribute to Medication Non-Adherence in Elders," *Generations*, vol. 35, no. 4, pp. 31–36, 2011.
- [14] B. Abbey *et al.*, "A remotely programmable smart pillbox for enhancing medication adherence." pp. 1–4, 2012.
- [15] C. Tandy and L. Bamford, "Medication compliance aids," *Eur. Geriatr. Med.*, vol. 1, no. 5, pp. 314–316, 2010.
- [16] "Nomad Concise MDS | Prescription2homes." [Online]. Available: <http://www.prescription2homes.com/nomad.html>. [Accessed: 12-Mar-2019].
- [17] "Monitored dosage system | MDS at a Glance | Venalink." [Online]. Available: <https://www.venalink.co.uk/mds/#mdsintro>. [Accessed: 12-Mar-2019].
- [18] "TabTime Medelert: Automatic Pill Dispenser with Solid Lid." [Online]. Available: <https://tabtime.com/products/tabtime-medelert>. [Accessed: 12-Mar-2019].
- [19] "Pivotell Advance Dispenser Pivotell Automatic Pill Dispenser, Alarmed Pill Box, Timed Pill Dispenser, Pill Dispenser." [Online]. Available: https://www.pivotell.co.uk/PivoTell+Advance+Dispenser/0_CAAA001/PRAA001.htm. [Accessed: 12-Mar-2019].
- [20] P. Weaver, *Success in your project : a guide to student system development projects*. Harlow, England : Prentice Hall Financial Times, 2004.
- [21] M. Sharma, "A study of SDLC to develop well engineered software," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 3, 2017.
- [22] A. Alshamrani and A. Bahattab, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model," *Int. J. Comput. Sci. Issues*, vol. 12, no. 1, pp. 106–111, 2015.
- [23] K. Petersen, C. Wohlin, and D. Baca, "The waterfall model in large-scale development," *10th Int. Conf. Prod. Softw. Process Improv.*, 2009.

- [24] S. Kumar, R. Rishi, and R. Rajkumar, "Spiral Increment Reuse (SIR) Software Model," *Int. J. Comput. Sci. Softw. Eng.*, vol. 5, no. 1, pp. 5–10, 2016.
- [25] A. Nilsson and T. Wilson, "Reflections on Barry W. Boehm's 'A spiral model of software development and enhancement,'" Bingley, 2012.
- [26] M. Stoica, B. Ghilic-MICU, M. Mircea, and C. Uscatu, "Analyzing Agile Development – from Waterfall Style to Scrumban," *Informatică Econ.*, vol. 20, no. 4, pp. 5–14, 2016.
- [27] B. Ghilic-Micu, M. Stoica, and M. Mircea, "Collaborative Environment and Agile Development," *Inform. Econ.*, vol. 18, no. 2, pp. 32–41, 2014.
- [28] R. Ramanujam and I. Lee, "Collaborative and competitive strategies for agile scrum development." pp. 123–127, 2011.
- [29] C. Pohl and H.-J. Hof, "Secure Scrum: Development of Secure Software with Scrum," 2015.
- [30] F. Siddiqui and M. Alam, "Ontology Based Feature Driven Development Life Cycle," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 207–212, 2012.
- [31] Z. Nawaz and F. Anwer, "Simplified FDD Process Model," *Int. J. Mod. Educ. Comput. Sci.*, vol. 9, no. 9, p. 53, 2017.
- [32] S. Palmer and J. Felsing, *A Practical Guide to Feature-Driven Development*. Coad Series, 2002.

Research Proposal

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

75/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

GRADING FORM: NEW ASSESSMENT

MATTHEW LEWINGTON**75****ENGAGEMENT WITH LITERATURE**

Good solid literature review that discusses the main user groups and their issues, together with existing systems. Well written with excellent use of the appropriate technical terminology; well referenced. A little on more on existing timed medication delivery would be useful.

20**KNOWLEDGE AND UNDERSTANDING**

Clear, SMART objectives with stated outcomes and deliverables. Clear deliverables, and a reasonable timescale. Good consideration of documentation to be presented as well as the physical prototype.

13**COGNITIVE, INTELLECTUAL SKILLS**



No need to describe a number of different methods – pick one and justify it. FDD is an excellent choice for a project like this, but there's a bit too much detail on the process for an initiatory document like this. Clear identification of the two separate data collection processes required for hardware/software and usability.

13

PRACTICAL SKILLS



Good choice of philosophy, approach and strategies. Good consideration of the choice between experiment and case study, particularly covering relevant ethical and regulatory issues. Would like a bit more on how the strengths of these will be used and how any weaknesses will be ameliorated. Remember to include people in your resources, both in terms of programmer and tester time. The Gantt chart is okay, but would be better showing the planned sprints/timeboxes of the FDD process.

17

TRANSFERABLE SKILLS



Good set of tables of stuff. Remember to centre tables and figures, and put the captions at the bottom of the display element. Check how you enter authors into Microsoft Word's referencing dialogue; the numerous initials means that they're not being picked up properly. You can't just copy and paste a line of authors, you must add them individually. Well-presented and well-ordered document.

12