Prototype driven development of information system:

A case study in Afghanistan industry

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Information Systems
in
Computer Science

Submitted by

Mohammad Azim Sakhizadah
Dawood Emran

Under the supervison of
Asst.Prof Rafiullah Mohmand

Information Systems Department
Computer Science Faculty
Kabul University
Kabul, Afghanistan

Auttum 2019

Abstract

Prototyping is the initial form of an actual system that can speed up the user interface design process and improve product quality, decrease the risk of building the wrong product. It is an initial model of the system showing the design and working of the system. Prototyping is a powerful method for the identification of usability problems at the very beginning of software development. Prototypes help connect the conceptual gap between a task model and a representational paper prototype for a user interface design. In this thesis, we have explained how the prototyping approach can help to successfully develop a management information system and software products, the steps that are needed to be taken to improve the prototyping approach and benefits of prototyping approach are discussed.

Contents

List of Figures

Chapter 1

Introduction

1.0.1 Background and Recent Research

Traditional systems development methods have become more complex and heavy, as the industry attempts to incorporate new tools in new applications. The authors believe that the old method - the old systems development paradigm - no longer fits the real world[Naumann and Jenkins, 1982]. In recent years, system developers have been searching for approaches that better adjust their designs with the needs, work practices and organizational settings of prospective users. Among such approaches, prototyping has appeared as one of the most likely[Trigg et al., 1991]. The building of prototypes has long been known as an important part of the software development process[Christie et al., 2012], and a prototype can be used to solve a variety of problems[Budde and Zullighoven, 1990]. Prototyping claims that decrease the risk of building the wrong product, through involvement and communication with the user. Also, prototyper claims to increase the quality of software products through a better understanding of requirements[Curtis et al., 1988]. So despite all the goodness of this new system, it is pretending to solve problems. It is also recognized as part of the Software Development Process. We want to research what is the most benefits of this approach. How we can promote this approach in software development companies in Afghanistan. In this thesis, we discuss the prototype driven development of information systems. In the first chapter of this thesis, we discuss our background and recent research(include of research topic, research question, research problem, research goals, research methodology, and research question). The second we bring our literature review and all of our studies. In chapter three, we discuss and show you the work that we have done. It Includes empirical surveys, conducting a workshop, data analysis, road-map development for prototype-driven development, assessment and evaluation. Finally, we have

a conclusion of all our work and our study.

1.0.2 Research Problem

Prototyping activities and tools have been recently gained great attention to product development research communities[Elverum et al., 2016]. There are many claims that prototyping known as part of the software development process. And also prototyping claims that decrease the risk of building the wrong product, through involvement and communication with the user. In addition, prototyper claims to increase the quality of software products through a better understanding of requirements[Phalp and Counsell, 2001]. Then prototyping when used correctly, the quality of the software will increase[Gordon and Bieman, 1993]. According to the importance of prototyping before this, there is no research about the prototyping, benefits, tools, and techniques of prototyping it in the context of Afghanistan. This study will address the tools, techniques, and benefits of prototype-driven development in the Afghanistan industry.

1.0.3 Impact of the Problem

Software projects are over budget, late and unpredictable. Sometimes the entire project fails before ever delivering an application[Reel, 1999]. The failure of software projects has many reasons. One of those is a misunderstanding of user requirements at the beginning of the projects. And we have heard a lot that collaboration helps us to implement the right requirements. But how clients and developers can achieve this collaboration? M. Schrage tells us about his experience that prototype and professionalism help us to achieve collaboration. Also, say that it doesnt go to a client meeting without prototype[Schrage, 2004].

1.0.4 Research Question

There are one main question with three sub questions that we would like to address with this thesis:
How the prototyping approach can help to successfully develop a management information system?

- What are the major benefits of prototyping-oriented MIS development?
- What steps are needed to be taken to improve the prototyping approach?
- What prototyping tools and techniques are needed to be considered to bridge the gap between theory and practice in MIS development?

1.0.5 Research Goals

To study the impact of prototyping in software development in the context of Afghan software development companies.

1.0.6 Research Methodology

The research method for this study is a literature review where we collect the actual case study reports for analysis. The sources will be used articles in conferences and journals, the papers should have appeared in a reviewed journal/conference and also technical resource and other theses. To complete our report we will find more resources through the internet news service. Finally, we will document our research.

1.0.7 Expected Outcome

Analytical report and a road-map for prototype-driven development.

Chapter 2

Literature Review

2.0.1 Prototyping Approach

Prototyping is defined as a preliminary working version of an entity or system. It is an initial model of the system showing the design and working of the system. The prototype of an entity or system can be in any form; it can be a paper-based(conceptual prototype) or working model(physical prototype) of a system. Conceptual prototypes are indicative in nature; whereas the physical prototypes are considered as either functional or simulated prototypes. Many writers view prototype as a throwaway (disposable) models of a system built as an experiment, which will be used to see and test the system, orient the users toward the use of the system, and encourage giving requirements from the user side. In some research articles prototyping is categorized as a *patched-up prototype, a nonoperational prototype, a first-of-a-series prototype, and a selected feature prototype.* All types of the prototype are evolutionary in nature because it is always desirable to alter the prototype and test repetitively.[Sajja, 2017]" Prototyping is recognized as a quick and repetitive development of a system by creating its operating model. It is an alternative to the systems development approaches like classical Systems Development Life Cycle (SDLC) approach and structured systems development (SSD it's conjointly called SSADM, Structured Systems Analysis and Design Method) approach. Unlike these 2 approaches, the prototype gives the addition of requirements and modifying the design at each iteration of the systems development method. Further, it's a fast method, wherever at the earliest, users will see the operating of the system. Because of its repetitive nature, the prototype model has become highly regarded. Besides prototyping, several connected approaches/models become widespread. One of them is rapid application development (RAD). RAD offers very little stress on typical analysis and design processes, however, it encourages the rapid

development of the working model. One like this approach is agile development. The agile methodology encourages the fast development of systems in very little parts by using XP(extreme programming) and scrum techniques. [Sajja, 2017]

2.0.2 Prototyping Steps

The first step of prototyping begins with the requirement gathering. In requirement gathering, both users and developers or system analysts are involved and document it formally. In the start point, the requirements of the systems clear the frame of the system staying to be developed. If exact requirements are known and identified well, the system will be made according to the requirements and may give a good quality system. However, good quality is the result of good system design.[Sajja, 2017]

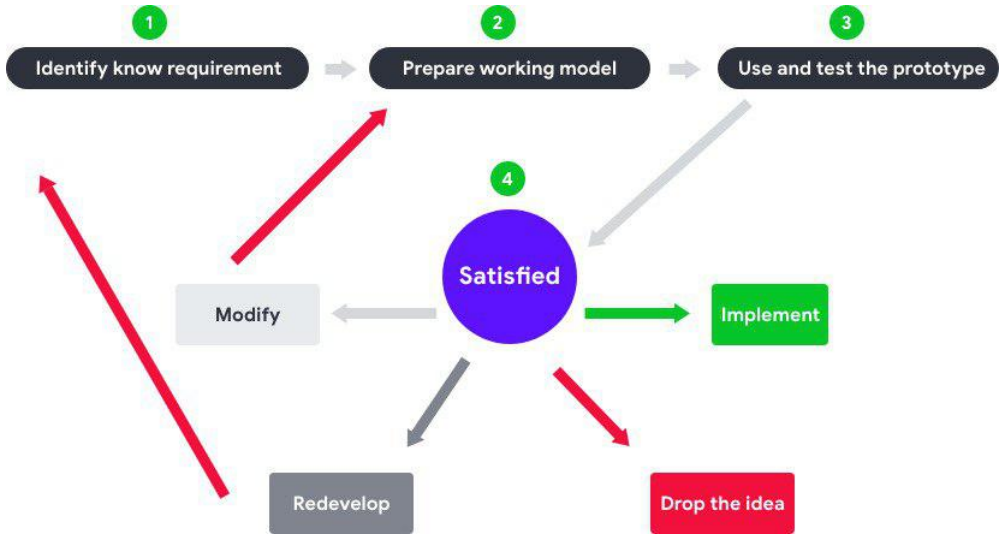

Figure 2.1: Prototyping steps

Figure 2.1 explain different steps of prototype approach. In the above figure 2.1 the first step is identify known requirements. In this step users and systems analysts both together identify the objectives of the system, selected functionalities, basic input/output, data required, and other requirements.

In figure 2.1 number 2 shows the second step of prototyping. In this step, you have to create a working model of the system using the collected requirements. If a system is being developed for non-computer professionals, then it may be desirable that just design the interface. Sometimes only the main functionalities of the system are simulated for the explanation. Usually,

the system out of any interface is first developed, especially when users are technically educated. There are tools accessible to develop the prototype of a system. Programming languages (such as fourth-generation programming languages), reporting tools, Computer-Aided Software Engineering (CASE) tools, etc. are rare common tools for prototyping. While developing the working model of the system, users are involved with the development process; nevertheless, the responsibility of the development is commonly on the Systems Analyst or the developer committee.

The number 3 of figure 2.1 shows the third step of prototyping. When the prototype created, the users take the lead role. The user should examine, evaluate and use the prototype following the direction of the System Analyst. Each user must judge the prototype according to their aspects and wants.

In figure 2.1, number 4 shows the final steps of prototyping that you must reexamine the prototype. If the developed working model(prototype) is accurate and satisfies every user. It should be performed directly. Although, no such complete prototype is made normally. The main cause is that the prototype has begun with well-known and available requirements, not all requirements. Furthermore, various prototypes are designed to extract more requirements by describing how the system would work. Another probability is to abandon the idea of the development of the system and throw the prototype. By developing the prototype, developers and users have examined different feasibility issues. If the result is not confident, and if it shows that the resulting the system would not be possible anyhow, the idea of developing a system is dropped. The next possibility is that the opinion of the development of the system is great, but not the prototype. Here a fresh prototype is required to be developed and tested again. This may be a result of any type of feasibility verification on the working model. The idea of developing a system is still good, but not with the working model approach. Usually, the prototype is satisfactory but not absolute. Examining and using a prototype opens up different requirements and shows various hidden expectations from the system. To develop the prototype according to the attached requirements and making it feasible, one has to alter the prototype. This method (examining and changing) is iterative in nature and remains until the satisfaction of users and developers.[Sajja, 2017]

2.0.3 Uses of Application Prototyping

After that prototyping shows how the proposed system would work and give a live presentation of the system before its original delivery, it is recognized as a blink test for the system. Whether the system would be received by

users, where it is operationally, economically, and technically possible, etc. can be examined with the cooperation of the prototype. It can be viewed as a requirement elicitation tool. By showing working of the system, users are encouraged to give more and more requirements. Several times, developers may not have sufficient domain knowledge; in that time including users in the development process warrants appropriate domain knowledge in the system. Moreover, users have trust in the system as well as the systems development process. Here are the better possible uses of prototyping[Sajja, 2017].

- As a systems development approach
- As a feasibility testing tool
- As a requirement elicitation tool

2.0.4 Strategies for Development of Prototype

Prototyping is a real sample of the system. It is not a complete system with all its expectations and requirements. First, the prototype starts with possible and recognized requirements of the system in the case to collect more extra requirements from its users and examine different feasibilities of the system. Several times it is important to present an environment and live presentation of the system, despite without developing the full-fledged system. In such circumstances, a partial working system will help the purpose. Such unfinished development may cover the development of front end interface(such as only screens), development of basic roles (without input and output interface, backup methods, etc.), and development of single processing functions such as input, process, and output. Figure 2.2 shows the major policies of prototype.

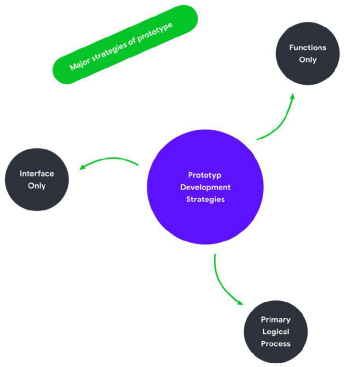

Figure 2.2: Major strategies of prototype

For those users that don't have pre experience of using such system, the design of front end interfaces would be very helpful. Through using such interface, users will feel good and begins accepting the system, still before its construction. Showing and using the screens and interfaces make a kind of trust among the users and affects the use of system. Users may give further requirements via using such screens. This approach works most suitable for end users who have less experience of computer-aided system.

The users, they are computer educated and associated in main procedures only, the strategy of developing main primary functions, would be proper. They are less involved in interface of the system. For such users the interfaces and input/output screens can be created later. Besides, backup service, database development, etc. can also be deleted. Alternatively, complete focus is given to chosen and essential functions of the system.

For some users who are engaged in studying the entire logical flow of the core system. For them main functions, input interfaces, and outputs of the system are produced as a prototype for more requirement elicitation and testing[Sajja, 2017].

2.0.5 Factors in a Prototyping Strategy:

1. *Prototypes can be of a single subsystem, of a set of subsystems, or of the entire system*

   While requesting the invention of a big system, it may be helpful to expand the work down into tinier subsystems that each of can be addressed with optimal strategy.

2. *Prototyping multiple concepts in parallel vs. prototyping only a single concept*

   finally, when only one or two concepts be chosen to fully developed, the development of various prototypes in an initial step can benefit to give valuable feedback.

3. *Iterative prototypes vs. only 1 prototype per concept*

   This portion examines whether or not it makes feeling to take a prototype all at once, or to focus on assuring that specific design requirements are satisfied before adding others. This is helpful when cutting down the number of concepts to a ultimate few, or interfacing working designs from various works.

4. *Prototypes can be virtual or physical*

Very complex analyses perform more easily by computer than by hand, and expansion of a computer aided design model can perform into product benefits for both prototype development and final product completion. Nevertheless, some of the most useful feedback a team can take for product development is to have a physical product available for user communication.

5. *Prototype manufacturing can be outsourced, rapid prototyped or completed in house*

   Outsourcing is resource significant in terms of price and time while not the item, however, it frees up the team to figure on different aspects of the project and provides access to resources that will not be on the market internally. Rapid prototyping technologies give quick production of elements for analysis, however on the market materials square measure restricted, which can not enable elements to be evaluated against design necessities. Finally, prototypes could also be completed in house, assume that skills and resources are available. this feature tends to be cheaper in price, however longer intensive for the team.

6. *Prototypes can be physically scaled*

   "With certain large products, such as ships and airplanes, creating a full size prototype may not be feasible until the final stage of prototyping, where any full-scale prototype is basically a final product. Additionally, for certain testing methods, such as wind tunnel testing, teams may not have equipment large enough to test a full size device."

7. *Prototypes can be functionally scaled*

   It is good for teams to create prototypes that include only a few design requirements at a moment, to be capable to correctly guarantee and estimate the successful implementation of inquired features. This can allow for more comfortable testing of prototypes and a more strong final product, but may point to issues when interfacing various prototypes into one final design.

8. *Prototypes can use similar or different materials than the final design*

   Because prototypes have the advantage of not having to meet final design requirements at all stages, some space may be given to material selection for prototype development.

9. *Prototypes can use similar or different manufacturing and assembly techniques than the final design*

By the start of rapid printing and rapid tooling, teams can determine if they want prototypes to be made and assembled similarly to the final products.

This list of factors is focused on the elements that contain significant engineering decisions about the progress from prototype concept to actuality. There are auxiliary portions, such as resource usage, that direct to traditionally fall more on the project management side. While these resource factors should be taken into account, it should be as an aide in deciding the importance of the primary prototyping factors. "[Christie et al., 2012]

2.0.6 Misconception for Prototype

Prototyping is a repetitive development process constantly try to help opportunities to modify them frequently. This causes that they impression that prototyping is casual and very trivial. The prototype is nor trivial neither casual, but it is a really serious work toward version-wise systems development with the scope of improvement at each repetition.

One another story about the prototype is that the prototype is only appropriate for very small and simple applications. At all, this is not true; prototyping is utilized when there is a high risk associated with the systems development process. Prototyping should be applied when the system is extremely important and a high degree of risk is associated with the system. The risk may be because of a big volume of data and system, complexity of the system, and the high price involved in the development of the system. If people are going for the way of their marriage suit (or dress), a system is far more important than that! Figure 2.3 illustrates the before-mentioned misconceptions.

Usually, the prototype becomes the individual duty of the developer (or Systems Analyst). Customers are not completely involved in using the working model. It is the responsibility of the users to evaluate the prototype and give property requirements and suggestions, from involving themselves in the overall development process.

2.0.7 Candidate Applications for the Prototype Approach

There are some applications that are new in terms of technology and procedures. At various times customers or users do not understand how accurately the system works, as they might not have observed such a system in practice. Such a new system with fewer known procedures and technology is the

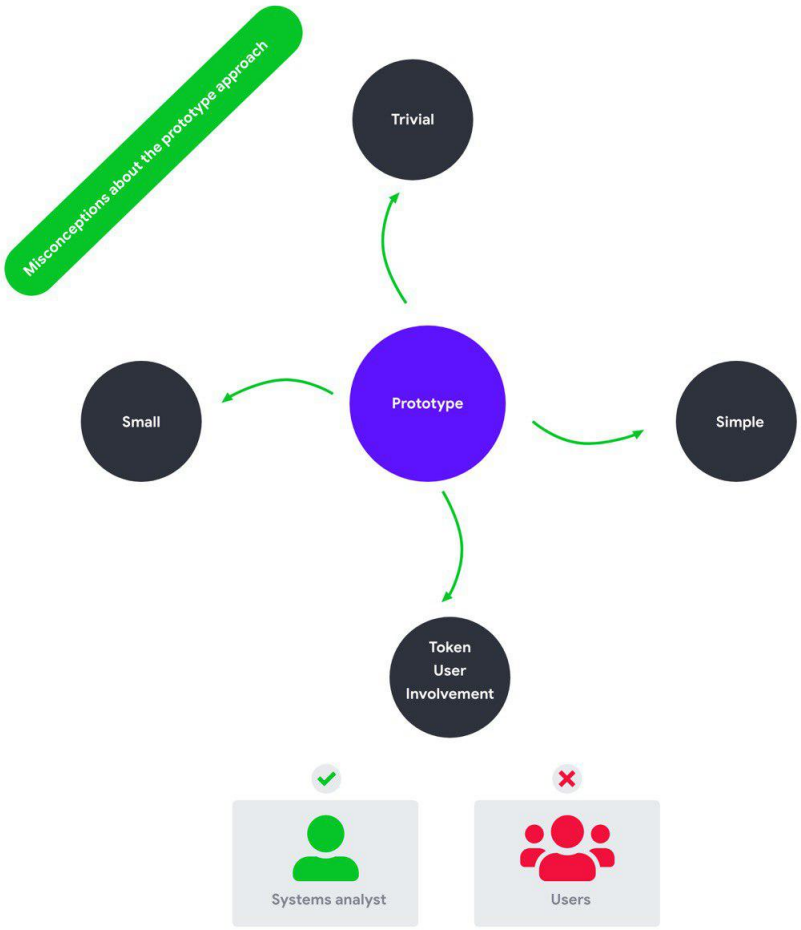

Figure 2.3: Misconceptions about the prototype approach

most suitable candidate for the prototype approach. It is more satisfying, to begin with, at least and known requirements and to develop a working model. Customers are connected at every step of the development, particularly at the evaluation stage, having their eye on the development procedure. By this way risk of using new technologies and innovative designed can be minimized. We can use this approach to observe and control the cost of the system also.

When an idea is new, to evaluate the usefulness of the idea, we prototype the system and evaluate it to confirm and certify many feasibilities of the system. Next, based on the usefulness evaluation performed on the prototype model of the system, the full-fledged system may be developed with either continuing the prototype approach or by selecting a new approach such as a

structured approach.

Similarly, when a full set of requirements is not formally recognized to the developers, the developers would like to give a presentation of the working system to extract more requirements by encouraging users.

To summarize, the following kinds of applications can be viewed as hopeful applications for the prototype development approach:

- *System incorporating novel technologies and procedures, which requires users approval.*
- *High-cost and high-risk type of systems.*
- *Requirements of the systems to be developed are not known.*
- *Systems that required to be checked for their feasibilities.*

2.0.8 Prototype advantages and benefits

The prototype is not suitable for every project, but it is right for the projects, it can be a great asset.

As Dr. Sheldon Cooper would say, when prototyping finds its ideal project match, its a bazinga winner.

Prototyping approach is a system development method(SDM) that a prototype is created, tested and revised as needed till an adequate prototype is finally achieved from which the entire system or product can be developed. A throwaway prototype made to know the project requirement before its design and coding begin. In reality, prototyping is a project test run [1]. Here are the advantages of prototyping. Let's start.

- *Improved communication among stakeholders:*

  We could not make the client engage in all projects like writing acceptance tests but prototypes did assist us to make sure that the communication we had was of better quality and that mutual understanding was reached faster. Schrage suggests that rather of a risk-adjusted collection of requirements, the main medium of communication between the development team and the client should be a prototype or model that would properly catch the real necessities of the client. It can usually be difficult to raise the amount of communication with the client and since reaching more excellent quality is one key to improved communication.[Käpyaho and Kauppinen, 2015]

[1] https://rapidsrepro.com/advantages-disadvantages-prototyping/

- *Improved and increased user involvement:[Cerpa and Verner, 1996]*

  Many clients want to feel like they are involved with the complex details of their projects. Prototyping wants user involvement and allows them to view and communicate with a working model of their project. With prototypes, clients can address their important feedback, offer project changes and change model specifications. The prototype most importantly supports eradicate misunderstandings and miscommunications throughout the development process.

- *Prototyping is widely recognized as one way of finding problems early: [Cerpa and Verner, 1996]*

  In the development of a particular application choosing the right and appropriate development approach is very critical. It can make the distinction that the project becomes success or failure. since the project advance, the cost of changing software rises exponentially. Boehm states that *finding and fixing a software problem after delivery is 100 times more expensive than finding and fixing it during the requirements and early design phases.* So experienced software developers mention here the usefulness of prototyping.

- The best approach that thought to improve communication between both users and IS personnel is a prototyping approach. This approach gives a very important role for customers during the software development process and this secures the best commitment form them. Information systems evaluated better by users and they feel more satisfied when the prototyping approach has been used[Cerpa and Verner, 1996].

- *Early validation of solutions and lower project risks[Passera et al., 2012]*

- *Thinking by doing, which enhances ideation, problem-solving and knowledge sharing between stakeholders[Passera et al., 2012]*

- *A stronger belief in the creative ability of the development team, and a sense of progressing forward[Passera et al., 2012]*

- *Product quality:* "In order to reach product quality, the prototyping approach is seen and understood as a suitable approach."[Nieveen, 1999]

- *Prototypes are also a means to communicate an idea to others:* A solid, obvious representation of a design concept is a shared view for all those involved in the design process."[Yang, 2005]

2.0.9 Prototype disadvantages

1. Reduced management control of project
2. False user expectation
3. Time required for user participation
4. Lack of necessary tools
5. Increase in development costs
6. Integration with existing methodologies
7. Inadequate analysis carried out
8. Poor documentation of systems

   The reference of all the above elements are available here in this paper[Kimmond, 1995].

2.0.10 Stages of prototyping

Also, prototyping may be realized by their stage in development. Sommerville classifies the prototype as incremental, evolutionary, or throwaway in software engineering. To bring changes in an existing product using the incremental prototype. An evolutionary prototype works into iterative steps of evaluation and building. Throwaway prototypes are an early step prototype that helps us to clarify the requirements. Petroski details the incremental path of several well known physical products. Budde, et al describes a classification system related to Sommerville, including evolutionary, experimental, and exploratory prototypes.[Yang, 2005]

2.0.11 Prototyping tools

In the world of design, one of the most popular phrases from IDEO is that *if a picture is worth a thousand words, a prototype is worth 1000 meetings.* Whether this is simple experimenting or changing your vision according to something tangible, prototypes help to build and testing while maintaining overall design flexibility.

As designers are obliged to design something that is explained to them by customers, not all actually know what the requirements actually are. This is later followed by a series of meetings, emails or phone conversations to reach their final design which really is a hassle. All these problems can be avoided by using prototyping tools. Prototyping tools enable designers and

clients to cooperate better while being in the same context rather than having different aspects. The customers get a visible summary of what is actually going to be made. It benefits teams to build understanding, to examine options and limitations that only become obvious when you build and test something. In the end, prototyping tools become a platform for full creativity and experimentation for the product team.

The most significant benefit of prototyping tools is the risk, the quicker we fail, the faster we would learn. When you fail with your mockup, you arrive softly theres always the opportunity to verify things that work, iterate and improve. Else, once we begin traveling on a boat in the wrong co-ordinates, we will never reach the shore.[2]

There are various tools today to assist you to produce digital products. They differ from developing tools to design tools to web creation tools. They are the tools that help you to make product creation quicker and much more useful. Prototypes express your ideas, and in doing so can change the way you design. Todays customers are seeing for interactive prototypes. These prototypes give you an overview of your design, interactions, and ideas. Repetitions and sharing abilities are really where prototypes stand out from the rest. The ability to work fast on different versions of your ideas and share them with your customers or your team is a real game-changer. In this section, we will present you prototyping tools which will help you design and iterate faster.[3]

1. **Sketch:** Sketch is a vector-based design tool which helps you design interfaces quickly and intuitively. Think of Photoshop in combination with Illustrator, but a lightweight software with unlimited artboards. Hundreds of articles have already been written on its various possibilities by industry experts, so I wouldnt list all of its features. Personally, one of its features that I like is nested symbols. With this feature, you can pre-design elements like buttons, widgets and even complex navigation mechanisms, then replicate their design on other artboards and shapes. Another feature that just came out is export to code by Launchpad for Sketch plugin which formats your design into code in seconds. It helps you get to development faster than ever before!

2. **Adobe XD:** Adobe has been known for its Photoshop and Illustrator. But last year Adobe launched a new UX prototyping tool Adobe Experience Design CC (formerly known as Project Comet). Adobe XD is

[2]https://medium.theuxblog.com/11-best-prototyping-tools-for-UI-UX-designers-how-to-choose-the-right-one-c5dc69720c47

[3]https://blog.prototypr.io/top-20-prototyping-tools-for-ui-and-ux-designers-2017-46d59be0b3a9

Figure 2.4: Sketch

focused around two tabs: Design, and Prototype. The Design tab features simple vector and text tools, and is used for creating your design. The Prototype tab is for previewing, and sharing your design. With Adobe, you can build high-fidelity prototypes also on a pc.

Figure 2.5: AdobeXd

3. **FIGMA:** Figma is a relatively new tool, with almost the same interface as Sketch. It is an innovative tool mainly because it allows a team of designers to collaborate and give comments on a design in real time. You heard right! Figmas canvas helps you do your design process in a collaborative way. Suddenly a very interesting situation occurs where the product team, the design team and the development team can take part in the process. Along with Slack, the entire team remains connected and the process could be fluid and efficient than ever.

Figure 2.6: Figma

4. **Axure:** Origami was initially created by Facebook to help teams build and design products. With this prototyping tool, we can preview the mockup live on our mobile in real time using Origami Live. Also, we can show off our designs in presentations in full screen, on a number of different devices.Sketch and Photoshop designs can be imported into Origami, and your project layers will be preserved, ready to be linked, animated and transformed as needed. You can also export your prototype components (including animations) with just one click,

so engineers can copy-and-paste into the project.One of the harshest drawbacks, though, is the lack of collaboration features. Theres little in the way of commenting and viewing version histories. This prototyping tool seems more attuned to freelancers or individuals just starting out in the business.

Figure 2.7: Axure

5. **Framer X:** Framer X has all the features you need to draw everything from custom icons to intricate illustrations. Fine-tune your designs with our advanced path editor, export anything from your canvas, and more. Advanced vector editing made simple. Draw everything from logos and icons to detailed illustrations with Framer X. Fine-tune your designs with our advanced path editor, create custom shapes with boolean operations, and export anything from your canvas with production-ready CSS and SVG code.

Figure 2.8: Framer X

6. **Mockplus iDoc:** Mockplus iDoc is a powerful product design collaboration tool for designers and engineers. It allows designers to automate design handoff and export design right from the Sketch, Photoshop or Adobe XD and see the specification prepared automatically. Create interactive and animated prototypes. Make design collaboration easier. Main features: Export designs in one click from Sketch, XD and PS Generate accurate specs, assets, code snippets automatically Show design tasks and workflow in full-view storyboard Comment right on designs to give instant feedback Build hi-fi interactive prototypes with real design files

7. **MockFlow:** MockFlow is an online UX suite for creative designers and usability engineers. its an easy to use tool to create wireframes and paper prototyping.

Figure 2.9: Mockplus iDoc

Figure 2.10: MockFlows

8. **Invision:** Among all popular prototyping tools, theres Invision, which also seamlessly integrates with Sketch. This is a Webby product that gives designers the freedom to design, review, test and share the results with developers and other team members. The most prominent advantage of this product, is its project collaboration features, which allow all users to give feedback, take notes and see the product changes in real time. InVision also offers a complete service for building prototypes for mobile, thus simulating the versatility of the digital product and its use in mobile.

Figure 2.11: Invision

9. **Atomic:** Atomic is a web-based tool, that requires Google Chrome. Since it does not have a desktop application its a drawback for developers using Firefox, Safari or any other browser. It gives you the flexibility and control you need to fine-tune your interaction: just click the play button to see your changes and animations in action. Atomic provides easy access to all developers by providing a shared prototyping system that is effortless. Hence, there is no need to download any app for collaboration. And the best feature of the tool is that is the history option, which allows you to rewind to see previous iterations and create new versions from any point.

Figure 2.12: Atomic

10. **Balsamiq Mockups:** Well end the list with another powerful prototyping tool called Balsamiq Mockups. It replicates the speed and convenience of creating mockups on paper, but on a digital medium. Designers can choose from more than 500 pre made icons and items or components they draw themselves. The interactions are basic, and the final, low-fidelity prototype feels more like an interactive, high-fidelity wireframe. If you are a UX newbie, this tool works wonders as it is highly straightforward to use. Balsamiqs sweet spot is the UX ideation phase. It produces really great rough sketches of the prototype for clients to view, which is a big plus for brainstorming sessions. Adopting really simple and fast keyboard shortcuts for users to increase productivity and the speed of creating mockups. With all of its low fidelity features, we can consider it more as a wireframing tool rather than a prototyping tool. If you are not a professional designer and you are looking for a tool that allows creating simple and static wireframes, Balsamiq can be used.

Figure 2.13: Balsamiq

In conclusion, these prototyping tools have their own special advantages, specialties and suit different needs. Admittedly, we have not included a lot of other prototyping tools for UI/UX designers. There are lots of other tools out there, most of which have the same features as mentioned in the above prototyping tools. Some honorable mentions are Moqups, UXPin, Prototype on Paper, Proto.io, and Flinto which have some unique features and ease of use as compared to the above. Ultimately, it all comes down to choosing the one that works best for you. If you are a seasoned designer working on complex animations, Principle might work best. Or, if you are a design newbie, then Balsamiq Mockups should be your go-to. We come from a time where mockups were made in paper and show to the clients, and now with this leap in technology we are capable of doing so much more. Going

beyond the boundaries of technology, down the line, we can expect to see more innovations and enhancements not just in the UX prototyping segment, but in the design industry collectively.

Chapter 3

Work Done

3.0.1 Conducting Workshops

At the start days of the research, we prepared our proposal. In the proposal, we defined a conducting workshops. During the research, we successfully did a workshop in our class. Here in the class, we presented our topic (**Prototype-driven development of information system**) to the students. What is prototyping? what is prototype-driven development and what are the benefits/ advantages and disadvantages of prototyping? Where and when use prototyping? We explained our topic and our goals from this research for the students to understand more about our research. Also, before the workshop, we had prepared an online questionnaire form for a case study and getting feedback's of our classmates to get actual data from the workshop. Finally, after the presentation we sent the form via email and Facebook group to all students to fill that. Fortunately, the form successfully filled out with some of our classmates. We wold like to say thank you all!. Here in this chapter, we will discuss more details of the workshop.

We built the questionnaire on Google Form, because Google form is a very useful and excellent tool to do the survey, case study and collect the data. Also, one of the greatest things that google form does, that it can analyze the data very beneficial and convenient, which is very helpful and excellent.

Now we are going to bring the charts and graphs and do analysis over them. Let's go on.... In the questionnaire we asked a multi chose question. What is your favorite field? (Web development, Database development, Android development, Designer, Software engineer) The reason we asked this question is to know which group of them in which field are familiar with prototyping. Is prototyping approach used in all fields? Is it helpful? In which field it is helpful? Figure 4.1 shows the favorite field of students that

they love and they are studying. The result as you see in the figure shows that prototyping used in all fields and is helpful for development of software products/systems.

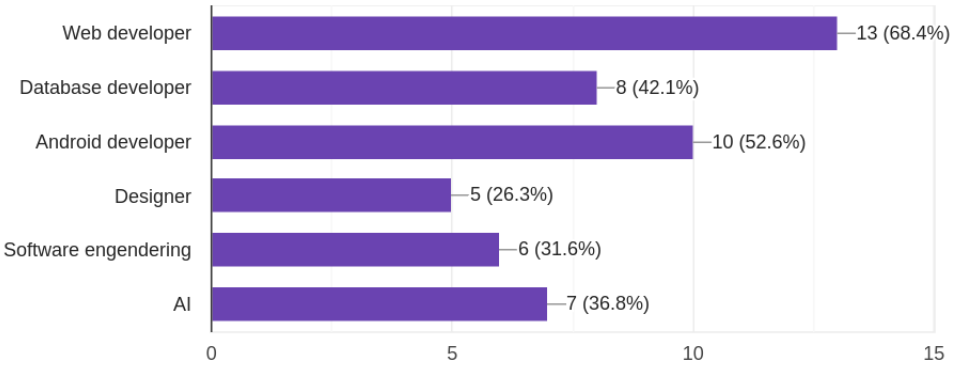

Figure 3.1: Favorite section

We asked, do you know about prototyping? Among 19 responses, 94.7% of them answered yes. they had know about prototyping. Figure 3.2 shows how many students know about the prototyping.

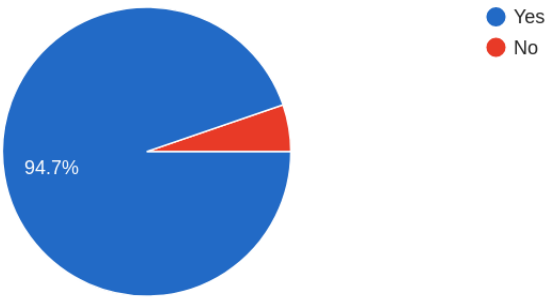

Figure 3.2: Do you know about prototyping

Figure 4.3 shows the answer to this question. Have you used prototyping? We asked this question because if they used prototyping and response to us yes, we direct them to a new section and ask some more questions about the prototype. If they replied no, we didn't ask them any questions more.

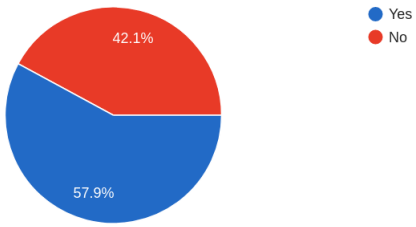

Figure 3.3: Have you used prototyping?

In the previous question, those who answered yes and used from prototyping we directed them to this question and asked them. If you used prototyping, how much is it helpful? Give rate from 1 to 5. Less value shows minimum help and high-value maximum help. As you see in the chart most of them give high score that yes prototyping help them.

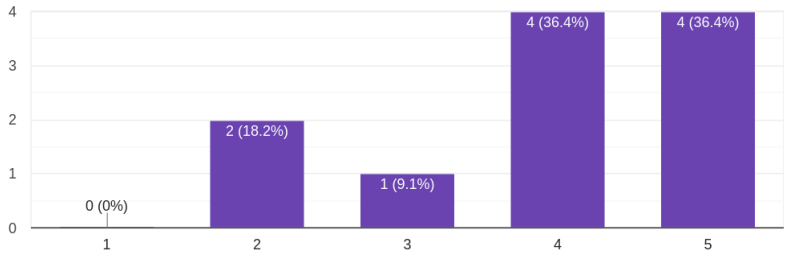

Figure 3.4: Does prototyping help you?

All questions are related to each other. This question is also made according to previous questions(Dose prototyping help you?). When they replied that they know about the prototype and used form this approach and yes it's helpful for them. We asked this question; In which part mostly prototyping help you? The question was a short answer question that they answered. We bring their responses and answers without any changes. In the chart, these elements were written by form fillers. They have written that prototype helps them in user understanding, a better understanding of user requirements, better project management, design and built the system. According to the below chart:

1. User understanding and Better project management have High score
2. Better understanding of user requirement has medium score

3. Design and built system has low score

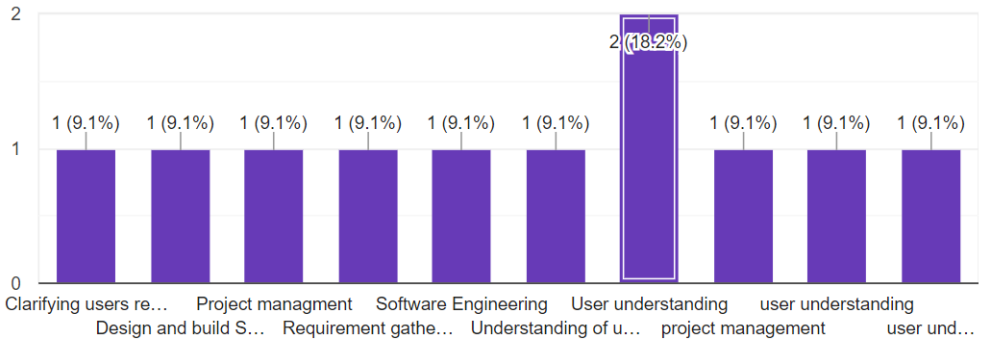

Figure 3.5: In which part mostly prototyping help you?

In the next question, we invited our classmates, that please share with us some information about the software development companies. If you are a developer or intern here. The reason to ask this question is that we wanted to know; are those companies used prototyping or no? They know about this approach or no? And if they used this approach how much is it essential in software development and MIS building? What techniques, methods, and tools have been used for prototyping? From 100%(totally 19 responses), 36.8%(7 response) was developer and intern in software companies and rest of them answer no they are not developer or intern. So from 7 responses, 2 responses said yes the companies that they are in, used the prototyping approach. Other 5 response said no the company that they are in, don't used this approach. So we can say that this approach is not commonly used in companies and don't become famous.

Next chart shows that do they accept prototyping as part of software development? There is no need to describe, see the exact result in the chart.

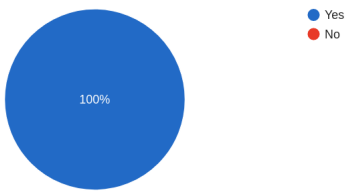

Figure 3.6: Do you accept prototyping as part of software development?

To find more exact results to our research questions, one of the questions we asked is: Is the prototyping approach helps to successfully develop a management information system? This question was a multi-choice question with these choices: strongly agree, agree, neutral, disagree or strongly disagree. So the result we get form this question is that 47.4% is agree, 42.1% strongly agree, and 10.5% is neutral that yes prototyping help us to successfully develop MIS's. In this question we don't have any response that show being disagree or strongly disagree. So we can say that prototyping approach is very helpful and great approach for building MIS's and systems.

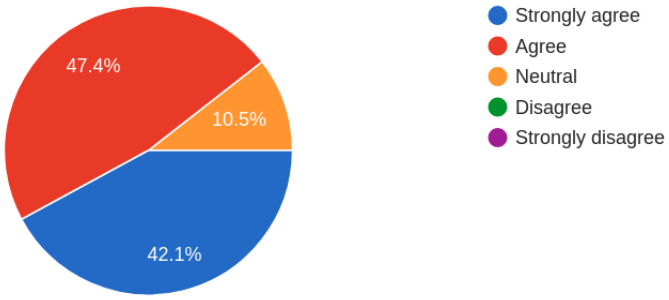

Figure 3.7: Is the prototyping approach helps to successfully develop a management information system?

Finally, we asked a short answer question that please write the most benefits of prototyping in software development? The below text is the exact answers of the participants without any changes.

- Reduce Time an Cost.
- User understanding.
- Improve and Increased user involvement.
- In agile method it can help for requirement gathering too.
- beginners developers also understand better requirement.
- The user misunderstanding will be solved and also the development time will reduce with users' feedback's.
- Prototyping is the first and the most important part of development and can lead a successful system.

Summary

We have done a workshop in our class and after that, we prepared an online form for a case study that successfully fills out by 19 persons of our classmates. Among all the participants that their favorite field sequence from top to bottom was web development, android development, database development, AI, software engineering, and designing 78.9% of them was a fourth-year student, 15.8% of them was a third-year student, and 5.3% of them was a second-year student. 94.7% of they knew about the prototyping, 59.7% of them used prototyping during the development of their project in university. Those who used prototyping give the score that 78.57% prototyping was helpful for them in these fields from top to down, user understanding, project management, clarifying user requirement and designing, and building of the system. 100% accept prototyping as part of the software development process, 94.7% recommend using the prototype, and 47.4% (agree), 42.1%(strongly agree), and 10.5%(neutral) are agree that prototyping approach help to successfully develop a management information system.

3.0.2 Empirical survey analysis

In our research, we had planned to have case studies in software development companies. To do that we tried several times in different companies for the case study and sent them an email. Unfortunately, they didn't accept us and even didn't reply to our email yes/no! (except Cyberaan software development company, that kindly helped us to do our case study here. We send our questionnaire for them who filled out by 10 developers of Cyberaan company). Its mentionable that doing research is a problem in front of researchers in software development companies/industries in Afghanistan. So after all trying, our teacher Prof.Ahmad Javid Baktash and Ast. Rafiullah Momand helped us to do our case study in mastery classes at Kabul university. Why mastery classes? Because here in these classes the majority of them were working in software development companies as a developer, designer, CEO, project manager, and software engineer. So it was a very good opportunity for us to do the survey. Fortunately, we successfully have done our case study over two class and get 23 response. we would like to say thanks to our teachers and all the participants that kindly helped us and filled out the survey paper. The lists of all companies that participated in our case study are: NMAA, ministry of interior, Wakhan software company, National Security Council of Afghanistan, National Examination Authority, Think Vision Group, AOP, OTTS, MUDL, FTP, AWCC, NOG, Netlinks(2 person), Ministry of education, NPA, General Directorate of, passport, Ghazanfar Bank, Independent directorate of local governance, Azizi Bank, First microfinance bank, Designing House, Cyberaan(10 person), and Pamir Elite. More analysis and details are in the next part.

In the questionnaire, we asked the experience of companies. How many years they have experience of software development? As you can see the exact result in the bellow chart that is like this: 1-3 year (57.6%), 3-6 year (30.3%), 6-10 year(3%) and 15+ year (9.1%). The chart shows that most of them have 1-3 years of experience. They are new and don't have a very long history. The experience of some of them is 3-6 years. It's also completely new and the next result also we get from this question that, in Afghanistan technology is new and doesn't have any long history. It is currently progressing in the new twenty years. And its use is expanding day by day. This is a great hope that our country becomes better very fast at the technological level.

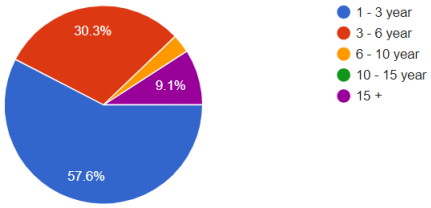

Figure 3.8: Experience of the companies

The next chart shows the methodologies that software development companies used in their system/software development. As shown in the chart, the methodologies used in these companies are: agile(the most used), waterfall(less than agile), and scrum(less used). This question is closely related to the next question we asked (Please write the steps/phases of the software development life cycle that you used). The reason we asked this question is that we wanted to know the methodologies and the steps of the software development life cycle that they used. Among the methodologies and steps, Do they have a step by the name of prototyping or no? Do they use the prototyping approach? Prototyping as part of software development process is as a part of software development life cycle or no? The question was a short answer question that we have been received 12 responses. Among all the responses, 3 responses have been write that they have used the prototyping approach in their development. Also they written the steps like this: requirement gathering, use case, design, prototype, development, testing, and implementation. So we can conclude that prototyping approach usage is very less and still it doesn't have most usage in product/system/software development in Afghanistan companies. Also most of them are not familiars

with this new approach and don't know about the benefits and techniques of prototyping.

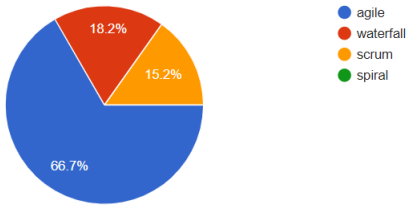

Figure 3.9: Methodology that used

Furthermore, we asked some questions, Do you accept the prototyping as part of the software process? Do you recommend using prototyping in software development? The answer to these questions shows that yes the majority of them(97% from total responses) accept prototype as part of the software development process and highly recommend using of prototype approach in software/MIS development. The next question is: Have you used prototype-driven development? From 33 responses in the answer to this question, 23 responses said yes they used prototype-driven development. 10 response said, no they don't use prototype-driven development. Finally, from the answers to the above questions, we can conclude that the majority of software development companies accept prototype as part of the software development process and the majority of them highly recommend using this approach. But still, this approach is not vastly in use and doesn't become stable in all companies. Some of the companies start prototype-driven development. Science this approach proposed instead of traditional development methodologies many researchers search about this approach give us good feedbacks and recommend this method. So we also try to improve this approach in the industries/companies in Afghanistan.

Most software projects completely or partially fail because they dont meet all their requirements. According to many studies, the failure rate of software projects ranges between 50% 80%. There are a variety of causes for software failures but the most common are Lack of user participation, changing requirements, inaccurate estimates of needed resources, badly defined system requirements, poor reporting of the projects status, poor communication among customers, developers, and users, poor Project Management and so on. We asked a question that, Is not using prototyping-approach, cause

project failure? As you see the result in the below chart, yes not using this approach can be one of the project failure causes. Because this approach helps both users and developers, it increases the involvement of the user, good tool for requirement elicitation, and it's good for project management and user understanding. Also in another response 36.4% of them are strongly agree and 51.5% of them are agree that prototyping approach help them to successfully develop management information systems.

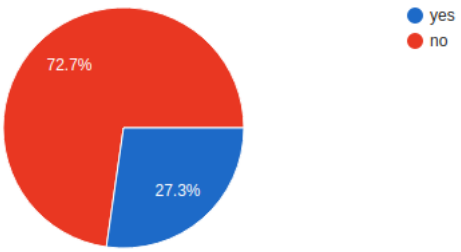

Figure 3.10: Project Failure Cause

Throw the literature review we found the most benefits of the prototyping approach. They include reducing costs of the software product, reduce delivery time, better project management, increase customer involvement, reduce project failure and, increase the quality of software products through a better understanding of requirements. So according to our study, we asked a question from our audience, that please rate each of them from 1 up to 10 that which one of them has the most benefits. Also, if you have some new options please add and rate it. The participants rated them and we analyzed all of them in a chart. As you see in the chart below the most one of the benefits is *increase the quality of software product through a better understanding of requirements*(high score) and *customer involvement*(high score). These two options have the best score and exactly these are right. As it's clear most projects fail because of poor user involvement and poor user requirement. When these two impotent and critical things are unclear and poor, the projects, systems and MIS's will fail. But prototyping helps both users and developers to communicate with each other to increase the quality of systems and decrease the failure of that. Also, other benefits are: reduce project failure, reduce the cost of the software product, better project management and reduce delivery time.

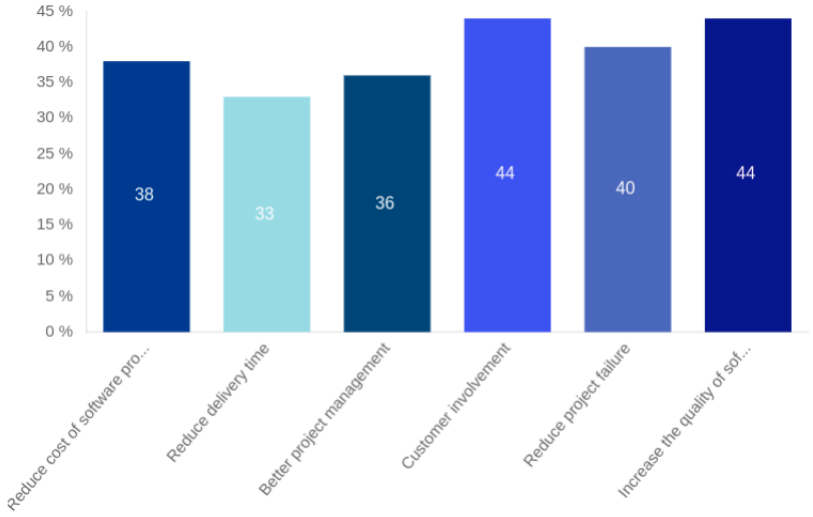

Figure 3.11: Benefits of prototype

Like the previous question, we list the most critical reasons for project failure. After that, we asked our form fillers to rate them that which one of them is the most cause of software project failure. The most causes of software project failure are vague requirements, communication breakdown, poor architecture, not using Prototype, poor user involvement, poor project management and many more. The participants rate them and we considered them in a chart. See the exact result in the chart.

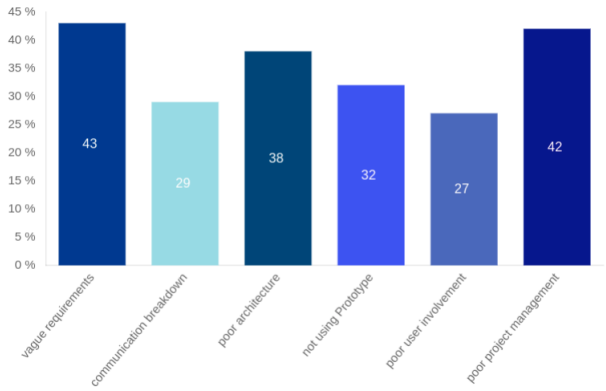

Figure 3.12: Project failure causes

The last tow questions that we asked in the survey paper are: If you used prototyping in MIS development, please write the tools and techniques you have used? What steps are needed to be taken to improve the prototyping approach? These questions was short answer questions that participants have been write some points here. In answer of first question they replayed just prototyping tools. They didn't write anything about the prototyping techniques. The tools that they used and give answer to our question is:

- Axure RP
- Sketch
- Origami studio
- Adobe XD
- Figma
- Webmarker and online softwares

Second question answers are:

1. At first we should know the customer requirement base on that we can base on that make the prototype.
2. making the prototyping as a culture for software development is a very good work to develop better softwares with high quality and according to user desires.

3.0.3 Roadmap development

The roadmap is a set of decisions or roadmap is a strategic plan that represents a goal or required outcome and introduces the major actions or events needed to reach it. Additionally, it works as a communication tool, a high-level document that supports clear strategic thinkingthe whybehind both the goal and the plan for going there. So, after all of our research, studying, workshop, and empirical survey we are going to propose a roadmap for prototype driven development of information system. The purpose of this roadmap is how to improve and implement prototyping approach in software development companies and industries in Afghanistan. Here we suggest the roadmap that includes these steps: change development management, change developers mindset, training and prototyping. Figure 3.13 shows the roadmap steps.

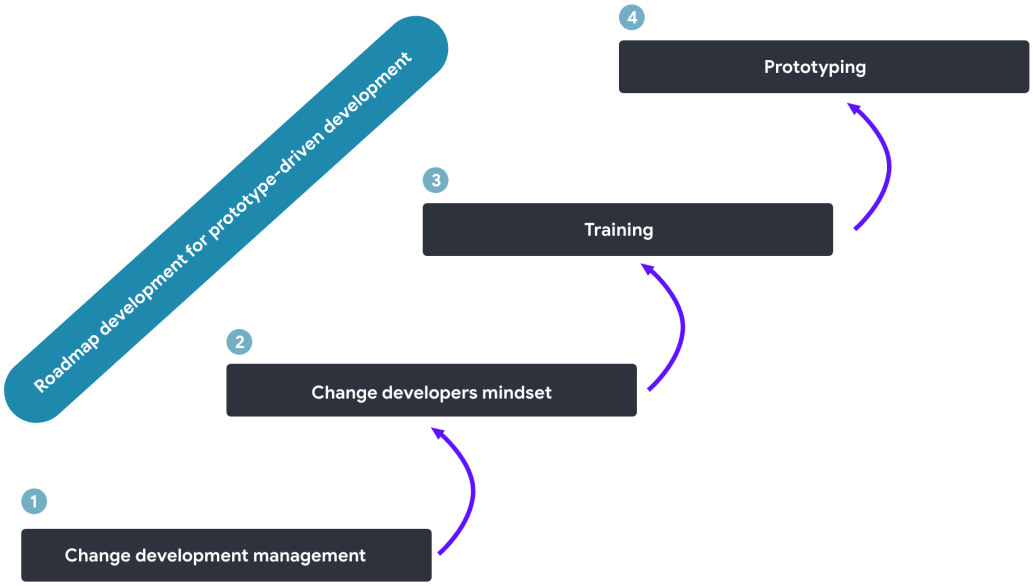

Figure 3.13: Roadmap

1. Change Development Management

   Every organization and companies have their strategies and management decisions and purposes. Software companies are not except for them. In software development companies also implement different strategies and methodologies. Policies may be internal and special for that company and methodologies are standards in all software development companies. There are many methodologies such as waterfall, agile, scrum, etc. All software development companies use one of these methodologies for the development of their systems and MIS's. So if some new changes and new things don't come, this law/policies and methodologies will be used continuously and don't bring any changes. But if some new changes, new laws/policies, new approaches come from outside and going to apply and implement in that company they must bring some changes to their management and policies to accept that or reject that. If they accept also they must search for the new approach and new policies to come in their development. They must evaluate all the side of that and predict some result that if they used and implement that approach how much changes come to their company and how much positive affect have over there. We propose here the prototyping approach to be implemented in software development companies. In the first step, they must decide to change the management of their company to accept new changes.

2. Change Developers Mindset

   All organization, companies and any kinds of place stands by human and they play the main role from every side. Some of them are CEO, some of them managers, decision makers, employees and sample workers. To implement new changes first of all, all the stakeholders must accept that and they must also changes themselves and be ready to adapt with new changes. Because they are the implementer of every changes and every tasks. Also software development companies have stakeholders or CEO, developers, designers, software engineers, decision mangers, manages and etc. They habit with technologies, methodologies, and techniques that used through development of systems and applications. To implement and bring the proposed approach in their company, all the stakeholders must change their habit, their attitude against the changes and become ready to accept new changes and adapt themselves with that.

   How bring changes to them?

   - describe prototyping approach: what is? when and how use?

- Benefits of prototyping.
- Showing of other feed backs and experience.
- Share good and ready material with them.
- Easy of use.
- Conducting some seminars and workshops.

3. Training

   When they successfully accept the changes and adapt them with that, this is time to start learning. Those whom going to use the prototyping approach they known from previous steps about: What is prototype? What is the benefits of prototype? Is it good to use? How much is it good? how much cause development of our company? and many more question. here in this step they have to choose an appropriate tools for doing prototyping. Because there are lots of tools and applications but among them we must discover one of the best. How to choose? Whit what factors they must choose? Let's discuss: When it comes to building prototypes, so various tools and techniques are out there that choosing one is no easy task. Which one is the best? Spoiler alert: There is no best because it all depends on what you need at the moment!

   1. How Easy Is The Tool To Learn?

   No design company or a UI, UX designer wants to spend hours learning how to use a complicated platform. The good news is that there are easy prototyping tools in the market that are straightforward and can be learned with zero troubles; the shorter the learning-cycle the better. In reality, most designers dont have the extra time to learn long-winded tools. In our view, a good start depends on how easy and intuitive user interface prototyping tool/platform can be taken.

   2. Does It Come With Support?

   Support is good. Difficulties and problems are inevitable, and having someone to talk to get help and support from is always important and a bonus. The quality and variety of support are also important. Are there help documents, guides, tutorials, and videos on offer and available to back such supports? This is a good question when thinking design prototyping tools based on support.

   3. How Long Does it Take to Create a Prototype?

   The best android prototyping tool and web prototyping tool factor in the actual time it takes when generating a real-life prototype using

uploaded image screens and other native or foreign tools. There is no reason why creating a prototype should not be quick and simple with all the choices that are available in the market.

4. Can you Test on Multiple Devices?

Does the tool/platform allow users to create a prototype for one device and test it out on all devices (ranging from the smartphone to tablet and a desktop)? This is a key requirement to determine the user experience across all devices. This allows users to quickly capture how their app looks and feels like across multiple devices before any complex coding can begin. In this way, it is quick and easy to change and improve designs before the app is built.

5. How Intuitive is The Tool?

How easy and well does the tool/platform allows users to simulate end-users intended interaction? This includes transitions, scrolling, gestures, custom transitions, navigation between screens and overall display and appearance.

6. How Easy is it to Share your Prototypes With Your Team and Clients?

The degree of features that is available on the design prototyping tools for sharing your prototypes with your team and clients. Are you limited to the number of people you can share your prototypes with? The rule of thumb in our opinion is to receive and take on board as much feedback as possible form each of prototype in order to refine and improve the final product.

7. How Easy is it to Collaborate on the Prototype?

Does the platform/tool allow teams and collaborators to easily communicate on the design screens? Having the freedom to invite, as many or as little collaborators to each of your prototypes is a plus point. Does it support real-time design collaboration? The ability to have your team and clients discuss, iterate and collaborate in real-time on a prototype upon the design screen is definitely a key feature to look out for.

8. Does It Allow Integration With Other Key Tools?

Teams and companies nowadays use so many other tools (e.g. Dropbox, Jira, Slack, Trello) in their everyday activity. Having a built-in integration with these third parties will help teams and businesses to automate many workflow processes involved. This will help in saving time and ensuring that every tool used represents the final version of the app.

9. Can you Keep Track of All Your Design Versions?

Feedback from your clients and team will surely spark creativity generating new design versions. Each feedback may provide new design version separately and maintaining these versions from every feedback can be very difficult to manage especially when you are working with large projects that deal with more than 50+ design screens.

Does the platform/tool provide a design repository to maintain all versions? A good question when evaluating and selecting a tool/platform. One should consider whether it has a built-in version history control panel that is quick and easy to view their evolving design files?

10. Pricing

This is last but by no means least; how much does the tool/platform cost to use? Some providers charge licence fees, yearly subscriptions, rolling month contracts and many more. All pricing plans are different and are all based on the number of users as well as the number of add-on features that they are offering. In our opinion, we feel the best two pricing plans are your rolling month contract. In this contract, you can cancel at any time as you are not signed up for a long contract. Especially when you feel that this particular one is not right for you. We also believe the pricing should increase the number of users/projects. In this way, customers are only paying for what they are using.

4. Prototyping Start the prototyping and develop your projects be leading of the prototype. **If a picture is worth a thousand words, a prototype is worth 1000 meetings.**

Chapter 4

Conclusion

The prototype defined as a primary working version of an object or system. It is the first image of the system showing the working and design of the system. The prototype of a system or entity can be of any kind; it can be a working model(physical prototype) of a system or paper-based(conceptual prototype). The steps of prototyping start with the requirement gathering. It is the first step of the prototyping that both the users and developers collect requirements and document them formally. The second step is building a working model from the collected requirements. The third step user takes the lead role and must examine, evaluate and use the prototype with the guidance of the developer. In the last step, you must review the working model or prototype and give feedback. If any changes wanted you must rebuild your prototype. Continues this step several times to complete your prototype and make satisfy your customer. The most desirable uses of the prototype are that it is a systems development approach, as a feasibility testing tool, and as a requirement elicitation tool. The prototyping approach has many benefits and advantages that increase the quality of software products. The major benefits of prototyping are Improved communication among stakeholders. This communication itself causes mutual understanding between users and developers or system analysts. The next benefit is that prototyping is widely recognized as one way of finding problems early. The next one is that by prototype we can early validate the solutions that cause lower project risks. Next is product quality: To achieve product quality, the prototyping approach is recognized and understood as a suitable and proper approach. Prototypes also mean to communicate an idea with others. Also for prototyping, there are many tools to help you deliver digital products. They are many different tools for design, for development, for web creation tools. Prototyping tools assist you to make your product creation quicker and much more useful. Prototypes show your ideas, and it can change the

way you are designing. Today our customers want interactive prototypes. These prototypes present you with an overview of your design, interactions, and ideas. Here in this thesis after the definition of research goals and research questions we searched more about the prototype approach and done case studies in some software development companies. Finally, we proposed a roadmap that how to improve and develop prototype-driven development in software development companies. The roadmap includes these four steps: change development management, change developers mindset, training and prototyping.

Acknowledgments

We are very pleased to end our research successfully as a final year project in computer science faculty at Kabul University. During this research in university we have learned how to make research. First, we would like to thank our advisor Prof.Baseer Ahmad Baheer at Kabul University, who helped us find the topic for this research. Thanks, dear Sir, we will never forget your assist, encourage, and your kindness. We wish you be always successful! We would also like to thank our supervisor Rafiullah Mommand to manage our project and help us within this research and guide us for providing the necessary tools which made it possible to complete our research. A bundle thanks to those who helped us in our case studies and fill out our questionnaire and attend our workshop. Finally, we would like to thank our family and friends for their moral support. Thank You

Mohammad Azim Sakhizadah
Dawood Emran

November 2019
Information Systems Department
Computer Science Faculty
Kabul University

References

[Budde and Zullighoven, 1990] Budde, R. and Zullighoven, H. (1990). Prototyping revisited. In *COMPEURO'90: Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering-Systems Engineering Aspects of Complex Computerized Systems*, pages 418–427. IEEE.

[Cerpa and Verner, 1996] Cerpa, N. and Verner, J. (1996). Prototyping: some new results. *Information and Software Technology*, 38(12):743–755.

[Christie et al., 2012] Christie, E. J., Jensen, D. D., Buckley, R. T., Menefee, D. A., Ziegler, K. K., Wood, K. L., and Crawford, R. H. (2012). Prototyping strategies: literature review and identification of critical variables. In *119th ASEE Annual Conference and Exposition*.

[Curtis et al., 1988] Curtis, B., Krasner, H., and Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287.

[Elverum et al., 2016] Elverum, C. W., Welo, T., and Tronvoll, S. (2016). Prototyping in new product development: Strategy considerations. *Procedia CIRP*, 50:117–122.

[Gordon and Bieman, 1993] Gordon, V. S. and Bieman, J. M. (1993). Reported effects of rapid prototyping on industrial software quality. *Software Quality Journal*, 2(2):93–108.

[Käpyaho and Kauppinen, 2015] Käpyaho, M. and Kauppinen, M. (2015). Agile requirements engineering with prototyping: A case study. In *2015 IEEE 23rd International requirements engineering conference (RE)*, pages 334–343. IEEE.

[Kimmond, 1995] Kimmond, R. (1995). Survey into the acceptance of prototyping in software development. In *Proceedings Sixth IEEE International Workshop on Rapid System Prototyping. Shortening the Path from Specification to Prototype*, pages 147–152. IEEE.

[Naumann and Jenkins, 1982] Naumann, J. D. and Jenkins, A. M. (1982). Prototyping: the new paradigm for systems development. *Mis Quarterly*, pages 29–44.

[Nieveen, 1999] Nieveen, N. (1999). Prototyping to reach product quality. In *Design approaches and tools in education and training*, pages 125–135. Springer.

[Passera et al., 2012] Passera, S., Kärkkäinen, H., and Maila, R. (2012). When, how, why prototyping? a practical framework for service development. In *ISPIM Conference Proceedings*, page 1. The International Society for Professional Innovation Management (ISPIM).

[Phalp and Counsell, 2001] Phalp, K. and Counsell, S. (2001). Coupling trends in industrial prototyping roles: An empirical investigation. *Software Quality Journal*, 9(4):223–240.

[Reel, 1999] Reel, J. S. (1999). Critical success factors in software projects. *IEEE software*, 16(3):18–23.

[Sajja, 2017] Sajja, P. S. (2017). Systems prototype approach. In *Essence of Systems Analysis and Design*, pages 81–92. Springer.

[Schrage, 2004] Schrage, M. (2004). Never go to a client meeting without a prototype [software prototyping]. *IEEE Software*, 21(2):42–45.

[Trigg et al., 1991] Trigg, R. H., Bødker, S., and Grønbæk, K. (1991). Open-ended interaction in cooperative prototyping: A video-based analysis. *DAIMI Report Series*, 20(365).

[Yang, 2005] Yang, M. C. (2005). A study of prototypes, design activity, and design outcome. *Design Studies*, 26(6):649–669.

Appendix A

Questionnaire

Prototype driven development of information system Prototyping is an initial instantiation of a concept as part of the product development process. It claims that decrease the risk of building the wrong product, through involvement and communication with the user. In addition, prototyper claims to increase the quality of software product through a better understanding of requirements. Then prototyping when used correctly, the quality of the software will increase. According to the importance of prototyping before this, there is no research about the prototyping, benefits, tools, and techniques of prototyping it in the context of Afghanistan. In this study will address the tools, techniques, and benefits of prototype-driven development in MIS development. Thus, this is a survey that will conduct by Mr. Mohamamd Azim Sakhizadah and Mr. Dawood Emran at Kabul University faculty of computer science as a final year project. You are invited to participate in this survey. The procedure involves filling a survey paper that will take approximately 10 minutes. The results of this study will be used in our thesis and journal paper and will be shared with any database.

Questions:

1. Name and Last name? ……..……
2. Email Address: ……..……
3. Level of education?
   a. Bachelor's degree. b. Master's degree. c. Doctorate. d. Professional.
4. Name your company/organization that you work their?

   ……..……
5. What is your position in the company/organization?
   a. CEO
   b. Software engineer
   c. Project Manager
   d. Software developer
   e. Designer
6. Experience of your company?
   a. 1 - 3 year
   b. 3 - 6 year
   c. 6 - 10 year
   d. 10 -15 year
   e. 15 +
7. Which method do you use in software development?
   a. Agile
   b. Waterfall
   c. Scrum
   d. spiral
8. Have you used prototype-driven development?
   a. Yes b, No
9. Do you accept the prototyping as part of the software process?
   a. Yes b, No
10. Do you recommend using prototyping in software development?
    a. Yes b, No
11. Is not using of prototyping-approach, cause project failure?

a. Yes b, No

12. Is the prototyping approach helps to successfully develop a management information system?
    a. Strongly agree
    b. Agree
    c. Neutral
    d. Disagree
    e. Strongly disagree
13. According to our study the bellow options are the benefits of prototyping (please rate them from 1 up to 10/ if you have some new option please add bellow in the blank space and rate that.)?
    a. Reduce costs of software product ☐
    b. Reduce delivery time ☐
    c. Better project management ☐
    d. Customer involvement ☐
    e. Reduce project failure ☐
    f. Increase the quality of software product through a better understanding of requirements ☐
    g. ……………………………………………………………………………………………… ……………………………………………………………………………………………… ………………………………………………………………………………………
14. According to our study these are the project failure causes (rate them from 1 up to 10/ if you have some other options please write bellow in the blank space and rate that)?
    a. Vague requirement ☐
    b. Communication breakdown ☐
    c. Poor architecture ☐
    d. Not using Prototype ☐
    e. Poor user involvement ☐
    f. Poor project management ☐
    g. ……………………………………………………………………………………………… ……………………………………………………………………………………………… ………………………………………………………………………………………
15. If you used prototyping in MIS development, please write the tools and techniques you have used?

    ……………………………………………………………………………………………… ……………………………………………………………………………………………… ………………………………………………………………………………
16. Please write the steps/phases of Software Development Life Cycle that you used.

    ……………………………………………………………………………………………… ……………………………………………………………………………………………… ………………………………………………………………………………………
17. What steps are needed to be taken to improve the prototyping approach?

    ……………………………………………………………………………………………… ……………………………………………………………………………………………… ………………………………………………………………………………………
18.

Appendix B

Workshop Survey Question

1. Name your company/organization?
2. What is your position in the company/organization?
    - CEO
    - Software engineer
    - Project Manager
    - Software developer
    - Designer
3. Experience of your company?
    - 1-3
    - 3-6
    - 6-10
    - 10-15
    - 15 +
4. Please describe the steps of your work during software development? (short answer)
5. Have you heard about prototype and prototyping process?
    - Yes
    - No
6. Have you used prototype-driven development?
    - Yes

- No

7. If you used prototyping, please write the tools and techniques you have used? (short answer)
8. What are the major benefits of prototyping?
    - Reduce costs of software product
    - Reduce delivery time
    - Better project management
    - Customer involvement
    - Reduce project failure
    - Increase the quality of software product through a better understanding of requirements
9. Do you accept the prototyping as part of the software process?
    - Yes
    - No
10. Do you recommend using prototyping in software development?
    - Yes
    - No
11. Is the prototyping approach helps to successfully develop a management information system?
    - Strongly agree
    - Agree
    - Neutral
    - Disagree
    - Strongly disagree
12. Which method do you use in software development?
    - Agile
    - waterfall
    - scrum
    - spiral
13. Project failure causes are?

- Vague requirement
- Communication breakdown
- Poor architecture
- Not using Prototype

14. Which one cause the most project failure?

    - Vague requirement
    - Communication breakdown
    - Poor architecture
    - Not using Prototype
    - Poor user involvement

15. What are the critical success factors in software projects? (shrot answer)