title: Homework Four and Five tags:

Tommy Arnzen

Due Date: Wednesday, November 15th, 2017

Assignment: Homework Four and Five

Class: Software Engineering

CSC540 Homework 4-5 (Human Factors)

- Deadline: Nov 15, 2017 (Wednesday)
- Submitted: Nov 15, 2017 (Wednesday)
- Time Analysis

	Warmup	Q1	Q2	Q3	Q4	*Total*
Estimation	120M	90M	90M	90M	180M	570M
Measurement	150M	95M	105M	100M	190M	640M
Date Completed	11/7/2017	11/8/2017	11/8/2017	11/9/2017	11/9/2017	11/7/2017-11/9/2017

1. Read a book and make summaries on technology factors/programming paradigm.

	Warmup	Reading One	Reading Two
Estimation	120 Minutes	90 Minutes	90 Minutes
Measurement	150 Minutes	95 Minutes	105 Minutes
Date Completed	11/7/2017	11/8/2017	11/8/2017

"Hackers and Painters" Chapter 13 - Revenge of the Nerds

- Summary Type: Thorough.

	Total Time
Estimation	90 Minutes
Measurement	95 Minutes
Date Completed	11/8/2017

Story

The chapter "Hackers and Painters" displayed an important concept in software engineering: The selection of a programming language can alleviate or add to the burden of a software project. An example can be seen in two organizations: organization A and organization B. Both organizations are developing a similar software system. Organization A decides to use a lesser known programming language (Lisp) to complete their project. They have determined that although this programming language is not as widely known as languages such as Java and C++ they have developers that can use the language to develop and maintain the system. Organization B decides to implement the software system using Java. The decision to use Java was based on the idea that Java is a popular language that meets the requirements of the software system. Management at organization B also believes that if the need presents itself they will be able to quickly replace developers on this project since Java is a popular language. After a period of time both organizations complete their systems.

Organization A's program is written in fewer lines, but organization B's program was able to implement more libraries and will be better able to be maintained over time.

The purpose of this story is to show that the selection of a programming language depends upon the organization that is developing the software system and the requirements of the software system's development. In the case of organization A it was decided that a lesser known programming language would be used to reduce the size of the program and enable the software system to be completed earlier than it would have been in other languages. In the case of organization B it was decided that Java would be used because it has more libraries, can be easily maintained over time, and developers can be easily replaced.

Different programming languages can be used to solve similar problems. In selecting a programming language an organization must take into account multiple factors before making a decision.

Arguments

- I agree that the selection of a programming language for a project should depend on the elements that the project is trying to address. The selection of a programming language should depend on the needs of the system.
- I agree that programming languages should not be selected based on how well they are known. It is important to be able to find programmers to write applications in the programming language selected, but this consideration should not be the only basis for a language selection.
- I believe that programming languages should be selected that can lead to the development of systems that can be maintained over time. I also believe that programming languages should be selected that are commonly known or easily teachable to new developers.
- I agree that programming languages should be selected that can help organizations avoid the mythical man-month that can result in the inability to add developers to long projects.

Takeaways

- Do not use popular programming languages only because they are considered to be industry standards.
- Programming languages are not all equivalent.
- Many organizations use popular programming languages such as C++ and Java because they are considered to be industry standards, and developers can be hired if the need presents itself.
- If programming languages were all the same then there would never be a change in industry standards. There would be no need to develop new programming languages.
- Java, Perl, Python, and Ruby are all similar to Lisp.
- The more powerful a programming language the shorter its programs will be compared to other programming languages.
- Programming languages that result in longer programs can take longer to develop, which can lead to problems in production.
- Languages vary in power.

"Hackers and Painters" - Chapter 14 - The Dream Language

- Summary Type: Thorough.

	Total Time
Estimation	90M
Measurement	105M
Date Completed	11/8/2017

Story

Person X has developed programming language Y. Language Y is a new mobile development language that was created to remove the burden of linking dynamic programming languages to XML user interfaces. Person X places the source code for the new language online where a small number of individuals find it. These individuals download the source code and begin writing test applications using the language's new user interface interaction features. As the individuals begin to see the

practicality of the new language compared to other options currently available they begin to show it to others. Soon there is a small group of users developing sample applications in the new language to test its functionality. Eventually person X begins to receive feedback, which is used to improve the language's main features, and enhance the reference material.

Eventually language Y's users begin to show their organizations the benefits that the new language can provide over their current methods. As the language begins to be used in commercial settings its benefits begin to be seen by more users. Over time the language becomes the standard programming language for a specific mobile operating system. Since the language is free, open sourced, and has a detailed collection of reference material it is continually used by developers of mobile applications.

Arguments

- I agree with the author that programming languages differ from software systems in that organizations will be less likely to use them if they are not free.
- I agree with the author that a popular programming language should be the scripting language of a system, but I also believe that the system must also be popular. In addition, I believe that as a programming language becomes more popular systems will begin to add support for it.
- I agree that a language should have some form of reference material. Reference material should not be frequently changed, and should be updated with new versions of the language.
- I also agree that a programming language should have examples of its functionality that are located in the same area as its reference material.

Takeaways

- Programming languages become popular as hackers start to use them.
- The popularity of programming languages also comes from their use in maintaining legacy software systems.
- Programming languages should be developed to serve a purpose.
- "A language has to be popular to be good." - Programming languages change over time as feedback is received from users.
- "For a programming language to become popular twenty programmers must decide to use the language on their own."
- Three requirements for a popular programming language - Free, offers a reference book, and is the main scripting language for a system.
- Programming languages should be designed to develop throw-away applications.
- Dream languages are open sourced and open design.

2. Watch a video and make a summary of factors/programming paradigms.

	Video Summary
Estimation	90M
Measurement	100M
Date Completed	11/9/2017

The Future of Programming - Robert Martin

- Summary Type: Thorough.

Story

Over the past seventy years computer science has seen a number of changes in many different areas. System elements such as hardware have increase in power and reduced in size, and demographics among software developers have fallen in age and shifted from a balance of males and females to a majority of males. These changes have been identified as the reason for many changes that have been seen in the software development process in recent years.

In the 1950s and 1960s a typical software development project would be completed by a group of individuals that included engineers, scientists, and mathematicians. Eventually organizations ran into the problem of finding individuals to complete software projects so they began to get workers from their planning and accounting departments. This resulted in software projects that were completed based on a process that was well defined and coordinated.

During the 1970s and 1980s the software engineering demographics began to change with a majority of developers being young males. This change in demographics can be attributed to a change in the nature of software engineering that became more focused on commercial development. During this period the waterfall effect of software development became common in which software systems were developed in an sequential fashion with design occurring before development. This resulted in projects that were not easily adjusted, which could make deadlines difficult to meet.

In the early 2000s the concept of agile programming was developed, which allows project requirements to more easily be identified, and deadlines to be more effectively met. Agile development involves identifying requirements through user stories, estimating completion times, test driven development, and collaboration with other developers and end users. The agile process was introduced to offer a method of completing projects in a way in which they could be easily adjusted, but at the same time completed in an coordinated manner.

Arguments

- I agree that programming languages should be well documented.
- I agree that software engineering demographics have changed because the nature of software systems and software development has changed.
- I thought that it was interesting when it was described that software engineering has not changed very much since the 1930s.
- I believe that many of the design elements that we have discussed can be applied to agile development processes without slowing them down. The application of design elements to systems developed using an agile process can increase the level of specification and understanding by reducing interleaving and introducing modularity.

Takeaways

- Programming languages should be developed for a purpose.
- Reference material should be short and well formatted.
- Programmers in the 1950s - Engineers, Scientists, and Mathematicians.
- Programmers in the 1960s - Experienced, Disciplined, and Professional.
- Programmers in the 1970s - Computer science graduates. Young and male.
- Agile developed from a need to manage programmers as the average age began to fall.
- Test driven development was originally seen in the 1950s and 1960s.
- The waterfall development process originated in the 1980s.
- Software has not changed very much since the 1930s.
- Most of the advances in software development since the 1930s involve identifying what not to do.
- Agile development requires developers to have discipline.
- Agile development involves estimating task completion times, communicating with customers, continuous integration, test driven development, and collaboration.
- Agile was described as discipline, craftsmanship, and professionalism.

3. Make a report on Agile process and answer questions.

	Agile Summary
Estimation	140M
Measurement	160M
Date Completed	11/9/2017

Agile Process

Executive Summary

The agile development process was developed in 2001 and has been used by many software development organizations to increase their overall efficiency. Agile software development aims at creating a process of continuous development, integration, feedback, and maintenance that allows user requirements to be met and systems to be enhanced over time. Agile development has a number of different methodologies, but the most common is scrum. Scrum development aims to meet user requirements by dividing projects in short sprints of development and individuals into different well defined roles. With our potential expansion into mobile development I believe that our teams should begin to look into implementing scrum and agile practices to enhance overall efficiency. The information contained in this report can be further referenced from the video "Intro to Scrum in Under Ten Minutes" and the article "Agile vs. Scrum vs. Waterfall." References to both of these articles can be found in the reference section below. The following is a summary of agile development methods as well as my recommendation for future development practices based on the positives and negatives of agile development and the organization's current development practices.

Current Development Process

- Currently our organization implements a waterfall process in development. Benefits that the organization gains from this process is that requirements can be identified in the design phase, changes in requirements do not frequently occur, developers do not require large amounts of management, and detailed documentation is created. (Agile vs. Scrum)
- Although there are benefits from our current development process there are also many disadvantages. The largest current disadvantage is that systems cannot be delivered until the late stages of projects. In addition, managers have trouble tracking overall progress, which can result in surprises and missed deadlines.
- An additional disadvantage is that although requirements are identified in the design phase of development the waterfall process does not allow changes to be made to requirements very easily once this stage has passed. If requirements are changed during development there is a risk that large amounts of the system will need to be rolled back to accommodate changes.

Scrum Development

Agile development can be described as a production process that allows developers to be more involved with end users through continuous interaction with stakeholders. The use of agile development allows constant updates and feedback from users, and allows for requirements to change during the design process. Different versions of agile handle interaction with end user differently. Scrum is one of the most common implementations of agile development. It is characterized by the use of individual roles that come together to form a team that is in continual communication with one another and with end users. Constant communication is one of the reasons that agile systems are found to meet customer specifications at higher rates than systems developed using other development methods.

User Roles and Scrum Process

In scrum development each team member is assigned a role. The major roles in scrum development are the scrum master, the product owner, developers, testers, and customers. (Intro to Scrum in Under Ten Minutes) The scrum master is responsible for coordinating the project. This includes setting up daily scrum meetings and monitoring the work down chart for each development group. Scrum meetings are short gatherings of 15 to 20 minutes that take place everyday, and allow developers to describe the work that they completed the day before, the work that they will complete on the current day, and any problems they encountered or expect to run into. These meetings give everyone present an idea of how far the system has developed, a general understanding of the system and how its components interact with one another, and can alert the scrum master to any deadlines that might be missed in the future.

The product owner is responsible for interacting with the software system's stakeholders. Based on the requests of the system's stakeholders the product owner communicates to the development team what the system should look like and how it should function. The product owner also works with the scrum master to decide which user stories should be given priority for the next release. The list of features desired by users in the next release is known as the sprint backlog.

Developers and testers in agile development work closely together through test driven development. Test driven development involves actively testing new features as they are being developed. This process allows unnecessary time for testing after development to be eliminated. By testing features as they are produced integration problems can be identified before additional features are built on top of them. This can eliminate the need for feature rollbacks.

Developers break their work into sprints which are small groups of functionality that are based on related user stories. User stories are the method that is used to describe user requirements in agile development. These stories allow requirements to be identified by viewing requested features from the point of view of the end user. Sprints typically take place over short periods of time such as a few days, a few weeks, or a month. If sprint time frames go over a month it is very likely that the user stories being looked at can be broken down into a series of smaller sprints.

After a developer completes part of a sprint they will update the work down chart. The work down chart contains a list of all active sprints that are currently being worked on. Work down charts allow the scrum master to track the progress of the development team, and reduce the likelihood that deadline surprises will take place. Based on the work that is completed over the course of multiple sprints the scrum master can begin to predict the level of production that a given development team will be able to complete during a certain timeframe. This allows the scrum master to make predictions for future deadlines.

Customers are the final group in the scrum development process. Scrum tries to involve customers in the development process as much as possible. Through the use of short iterations of development prototypes can be developed that allow the development team to get feedback from customers at every stage of the development process. The interaction between the development team and customers takes place through the product owner.

The agile development style offers many benefits to developers and end users. The following is a list of potential benefits that could be gained from implementing a scrum process.

Advantages

- "More transparency and project visibility" (Agile vs. Scrum) - By holding daily meetings the scrum master is able to assess the overall progress of the software system on a consistent basis. In addition, through the use of daily meetings a greater understanding of the system can be gained by having each member explain the context that their portions of the system add to the overall functionality.
- "Easy to make changes" (Agile vs. Scrum) - Since customers and end users are involved in the development process at every stage feedback can be gained on features as they are implemented. The process of receiving feedback at every stage of a project allows problems to be identified before the system is completed.
- "Reduction in costs" (Agile vs. Scrum) - In agile development cost reductions can be gained by identifying problems before they require costly rollbacks. Through the use of small teams and customer feedback problems are more likely to be identified earlier than in waterfall style projects where customers do not see the final system until it is completed.

Although the agile development process offers many benefits in software development it can also present problems.

Disadvantages

- "A lack of documentation" (Agile vs. Scrum) - The sprint development process can result in focus being placed more on functionality than on producing documentation. This can be a problem because as features are created their functionality and interaction with the system can be overlooked. In addition, the lack of detailed documentation can result in an inability of developers to maintain the system in the future, and an inability of user to understand how to use the system.
- "Planning can be less detailed" (Agile vs. Scrum) - Since agile development is based on identifying user needs and creating sprints to meet those needs developers can begin to focus on individual system elements rather than on overall system functionality. This can result in developers overlooking the requirements needed to test the finished system.
- "Set design is not often present" (Agile vs. Scrum) - Since end users are involved in every stage of the development process their requirements can change as the system is being created. This can result in a final system that does not resemble the original system design. This can be a problem if project deadlines are set based on the original requirements.

An overview of the agile development cycle can be described as follows:

Development Lifecycle

- Identify Requirements.
- Plan Development.
- Design System.
- Develop System.
- Release System.
- Maintain System.

Conclusion

Describe Agile Development Process

The current waterfall development process that has been in place at our organization for the past decade has been an effective method for producing general purpose database systems, but I believe that with the potential entry into mobile development a change should be made to move towards an agile form of development. This does not mean that database software teams necessarily need to change their development style, but mobile development teams should begin to use agile processes to increase efficiency, reduce management problems, and react to changes more effectively.

References

- "What's the Difference? Agile vs Scrum vs Waterfall vs Kanban." Smartsheet, SmartSheet, 3 Nov. 2017, www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban.
- "Intro to Scrum in Under Ten Minutes." Youtube, Youtube, www.youtube.com/watch?v=XU0llRltyFM&t=7s.

Q & A

	Agile Questions and Answers
Estimation	40M
Measurement	30M
Date Completed	11/9/2017

Q1. What are the differences between the agile and scrum processes?

Agile development is a software development process that allows software systems to be completed quickly and in an efficient manner. Agile development typically uses sprints, which allow features to be identified and completed over the course of short periods of time. During agile development it is common for test driven development to occur. Test driven development is the process of testing new features as they developed.

Scrum is a version of agile development that focuses on roles that define processes that need to be completed over the course of a project. Through the use of roles each individual working on a project has a specified task to complete. By identifying roles expectations can be clearly communicated, which can allow the development process to be more efficiently carried out.

Q2. Why is the agile process popular? Also, why do some people dislike it so much?

- One reason that the agile development process is popular is because it allows projects to be completed with a higher probability of meeting deadlines. The agile development process also allows work to be completed in a manner that allows for unit testing during development, and updates to be made based on user feedback and communication. Through the use of progress tracking elements such as work down charts and daily meetings surprises are reduced in the development process.
- Reasons that individuals do not like the agile process include a perceived lack of system design and architecture, and an over emphasis on short term development that can take focus off of the overall system and reduce system

understanding. If an individual only develops small portions of different system elements the result can be a lack of an overall understanding of how the elements they design fit into the functionality of the entire system.
- Another reason that some individuals dislike agile development is that it can result in a lack of detailed documentation. Since focus is place on development through short-term sprints documentation can be overlooked.

Q3. How is the agile process executed? What should we expect from the execution of the agile process?

- The first stage in the agile process is to designate individuals to specific roles. Defining roles at the beginning of a project allows project members to understand the tasks they are supposed to complete. The main roles that individuals will be assigned are project manager, product owner, developers, testers, and customers. For each project there will be one project manager and one product owner.
- The responsibility of the project manager is to coordinate the project through the use of short daily meetings, assign individual roles, and monitor work down charts to make sure that the project is progressing as expected.
- After roles have been assigned user stories are identified. User stories are used to describe needed system functionality form the point of view of the end user. After a list of user stories has been identified similar functionality is grouped together to be developed in sprints. Sprints are short periods of time during which certain features are developed and tested. Each user story is assigned a predicted time value that developers believe it will take to complete. Each sprint along with their individual user stories and predicted completion times are added to the work down chart to allow their progress to be monitored over time.
- One of the main benefits of agile development is that changes can be made to the project at every stage. If it is found that the completion time of a sprint was not predicted correctly the sprit can be adjusted to meet deadlines.
- Once development begins progress is monitored through the use of work down chart updates and daily progress meetings.
- If the agile process is adopted the areas that will see enhancements are project coordination, a better ability to make adjustments to changing customer requirements, a higher ability to monitor project progress, and a better ability to avoid and deal with problems that arise in the development process.

Q4. What are the pros and cons of the agile process?

Pros

- The ability to make changes during the course of a software project is introduced because the system is not being developed entirely at once. Since the system is developed incrementally changes can be made at different stages of development without having a large effect on deadlines.
- By identifying and specifying user needs and system components time estimations can be made that can allow deadlines to be met more constantly than other development approaches.
- Agile development allows consistent interaction and feedback from end users. This allows for a higher probability that the end product will be what the user desires.
- Feedback can also be gained from other team members in the form of quick daily meetings and work down chart updates.

Cons

- The use of agile development can lead to a lack of overall system understanding if focus is placed too heavily on individual tasks rather than on their interaction with the entire system.
- Agile development can result in a lack of initial design and architecture planning. This can lead to problems of interleaving and non-modularity during development.
- In agile development documentation is not always recorded as well as it might be under other development methods. Since system aspects are developed in short sprints this can lead to an unclear understanding of the system after a couple of sprints have been completed.
- One of the main benefits of agile development is that it can be used to clearly define user requirements, but the ability of agile to allow for constant changes during development can lead to systems that do not fully represent the initial needs of users. If a system is changed from its original purpose during the development process the benefits gained from the ability to clearly define user requirements at the beginning of a project can be lost.

Q5. My team has 10 members and mainly does software reliability testing, do you think my team should adopt an agile process?

- Agile development is often used with test-driven development in which system components are tested as they are developed. As a result agile development methods could be used by testing teams. For a testing team the agile process would be changed to focus more on sprints dealing with testing functionality rather than developing elements to meet user requirements. Using work down charts would only be effective if new test cases were not expected to be introduced at varying times. Rather than tracking the development progress of the testing team work down charts could be used to predict the number of test cases the team can complete during a specified period of time. These predictions could then be used to determine the amount of test cases that can be processed by the team and the amount of time those cases would take to run.

Q6. We formed a new team of five members to explore a new product, do you recommend my team to adopt an agile process?

- I believe that the decision to use agile development methods should depend on the organization developing the software system and the problems that the system aims to solve. An agile process could be applied to this situation that would allow user needs to be identified through user stories and development to occur in a coordinated fashion. If the team has never used an agile process before I would not recommend using every aspect of agile programming. I would recommend that the team use user stories, sprints, and daily progress meetings to track their progress. I would also recommend that the team limit the role of the product owner until they are familiar with the agile development process. The use of a product owner could result in multiple changes being requested during development, which might introduce problems in production for a team that is not familiar agile development.

Q7. How can the agile process be used to identify needed system elements?

- In the agile development process there are a number of ways that needed system elements can be identified. The first method is user stories. User stories allow system requirements to be identified from the point of view of the system's users. User stories are then placed into a work down chart of items that are to be completed. The second method by which needed elements can be identified is through daily scrum meetings in which developers meet for a couple of minutes each day to discuss the work they completed the past day, the work they will complete on the current day, and any problems they expect to run into. This stage also allows changes to be made to the project as it is in progress if efficient development methods are discovered during the course of the project. The final method for identifying needed system components is through unit testing. Through the process of running unit tests elements can be discovered that need to be changed or added to the system to carry out desired functionality.

Q8. How can the agile process monitor the progress of a project over a period of time?

- In agile development project progress is monitored through the use of work down charts. At the beginning of a project user stories for system requirements are selected and placed in a work down chart. The work down chart contains each element that needs to be added to the system as well as the estimated amount of time in hours, days, or weeks that the element is expected to take to complete. As different elements of the system are completed the work down chart is updated to show the project's progress. Over time different sprints can be compared to one another to see if the project is on schedule. The work down chart reduces the possibility of surprises in the software development process because it displays the amount of work that has been completed and the amount of work that needs to be completed. In addition to the work down chart daily scrum meetings can be used to identify if certain system aspects have been completed and if problems have been encountered.

Q9. What are the main team roles in the agile process?

- Product Owner - This individual makes sure that the right features make it into the product. In addition, the project owner is also responsible for communicating with the end users.
- Project Leader - This individual makes sure that the project is progressing smoothly. The project leader holds daily scrum meetings, monitors work down charts, and makes adjustments over the course of a project to increase the productivity of the project developers.
- Developers - These individuals develop the product.

- Testers - These individuals test the product.
- Customers - These individuals user the product and offer feedback to enhance the software system.

Q10. How is progress communicated during the course of a project, and are changes made to schedules during the development process?

- In agile development progress is communicated by developers over the course of a project through updating work down charts and giving updates at daily scrum meetings. Progress is communicated to end users through prototypes presented by the product owner. The product owner is responsible for making sure that the correct features make it into the software system. To ensure that the correct features are developed and that they meet customer specifications product owners communicate with end users at every stage of the development process to give updates and get feedback.