Waterfall: Clear System Design

The main objective of the waterfall model is to have all the requirements analyzed and cleared upfront, to prevent the development of software going downhill without the possibility of making changes. One of the problems that waterfall is trying to address is clear system design; it states, “If the documentation is bad design is bad.”[1] After having a clear requirement from the requirement documentation engineers will be able to:

“design the overall architecture of software to fulfill software and hardware requirements.”[2]

Essential:

Problem with difficulty in designing clear system is a problem with complexity. As the requirement for software grows so does the complexity of the system and the visibility of software. By definition, more features mean more tangible system, which is unavoidable. For this reason, this goal of waterfall method is trying to solve the essential problem.

[1] W. Royce, "Managing the Development of Large Software Systems," *Proceedings of the IEEE WESCON*, pp. 328-338, Aug. 1970   
[Online] Available: [http://dl.acm.org/citation.cfm?id=41801](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/Royce_waterfall.pdf)<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>

[2] “What is waterfall methodology,” *Eastern Peak*. [Online]. Available: https://easternpeak.com/definition/waterfall-development/. [Accessed: 17-Nov-2019].

Rapid Prototype: Extracting user needs

The main objective of a Rapid prototype is to have something to show to your customer before you start making the whole system. Prototyping “helps in extracting user needs, which are presented, and defined by building a working model.”[1] The main goal is to get the final sense of design of customer need.

Essential:

Human is not very good at expressing and describing things. Sometimes we cannot describe the things which go into our mind. As said by the author of SE-III course, “Communication problems are essential because all humans have an inherent difficulty expressing and understanding complex ideas.”[1] With the help of prototype customers can be able to explain what he needs and what he needs not in his final software. For this reason, extracting user needs is essential problem.

[1] J. Carey, "Prototyping: Alternative Systems Development Methodology, "*Information and Software Technology*, vol. 32, no. 2, pp. 119–126, Mar. 1990,   
[Online] Available:[http://www.sciencedirect.com/science/article/pii/0950584990901114](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/Prototyping_alternative%20systems.pdf)

Spiral: Project Risk

The main objective of the spiral method is to handle all the unknown risks that come while the project is running. The spiral method believes in “learning with maturity which involves minimum risks for the customer as well as the developmental firms”[2]. Spiral methods choose the best solution out of all the alternative solutions proposed for the given risk.

Accidental:

Most of the time project risks are prices, performance, workstation cost, laws for implementing certain things in software, etc. These are inherited in software development; these come based on where we are and what resources we have. Therefore, the project risk problem is accidental.

[1] B. Boehm, "A Spiral Model of Software Development and Enhancement," *IEEE*, xxx, pp. 61–72, May 1988,   
[Online] Available: [http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=59&tag=1](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/A%20Spiral%20Model%20of%20Software%20Dev.pdf)

[2] “SDLC - Spiral Model,” *Tutorialspoint*. [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc\_spiral\_model.htm. [Accessed: 17-Nov-2019].

Agile- XP: Changing requirement

The main feature of the Agile-XP is to adapt the human nature in software development. It addresses the “dynamically changing software requirement.”[1] It believes that customers may not know what they want at the initial stage of the project.

Essential:

Problems with changing requirements are actually the problem with changeability. Humans have a hard time imagining things if they have never seen or experienced such things before. The same comes with the requirement, it is hard to tell one will need in project at upfront to make project successful. Therefore, changing requirement problem is essential.

[1] “What is Extreme Programming (XP)?,” *Agile Alliance*, 24-Sep-2019. [Online]. Available: https://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'aa\_book~'aa\_event\_session~'aa\_experience\_report~'aa\_glossary~'aa\_research\_paper~'aa\_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1). [Accessed: 17-Nov-2019].

Agile- Scrum: Team-Work:

The main objective of Scrum is to deliver business value in the shortest time by focusing on accountability, teamwork, and iterative process towards a well-defined goal. It uses:

“one or more cross-functional, self-organizing teams of about seven people each.”[1]

Accidental:

Teamwork is something that can be improved with training and good management. Using of good framework like scrum helps to unify team of different disciplines and makes software development more coherent. This is something that is not inherited problem of software development but rather more towards how the team is formed and what their common goal is, and it is fixable. Therefore, the team-work problem is accidental.

[1] M. James, "Scrum Reference Card," [Online] Available:[http://scrumreferencecard.com/ScrumReferenceCard.pdf](https://content.byui.edu/items/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/?.vi=file&attachment.uuid=e38e0945-caed-4675-a65f-8eeb24eeb4ff)

Cleanroom: Defect prevention

The main objective of the cleanroom process is the development of software that exhibits zero failure. It was created for hardware industry where even a single error could be very dangerous. It combines statistical analysis methods of software specification, design, and correctness verification. In other words, it focused on “defect prevention rather than defect removal.”[1]

Accidental:

Defect prevention is something that is focused on minimizing or eliminating human mistakes during software development. It can be a small typo error or can be variation in units of measurement. But these types are problems can be solved and are mostly caused by human negligence. Therefore, defect prevention problem is accidental problem.

[1] R. Linger and C. Trammell, “Cleanroom Software Model,” *CMU*, Nov-1996. [Online]. Available: https://resources.sei.cmu.edu/asset\_files/TechnicalReport/1996\_005\_001\_16502.pdf. [Accessed: 17-Nov-2019].

Formal Methods: Testing

The main objective of the formal method is that it helps test software implementation in every phase of software development. It makes sure that: “software satisfies the specification.”[1]

Essential:   
Testing is one of the inherited problems in software development. A lot of resources are spent in testing the software before it can be released to customer. No matter how much good design or implementation was; it is mandatory to do unit testing, quality testing, etc for successful software development. Therefore, testing problems is essential.

[1] *Role of Formal Methods*. [Online]. Available: http://homepage.cs.uiowa.edu/~fleck/role.html. [Accessed: 17-Nov-2019].

CMM: maturity leveling

The main objective of CMM is to Level companies based on their “capabilities and quality.”[1] The aim behind maturity is it will help customers to find the right company for their product development, and it will help companies to see where they are and how can they optimize to go for higher level.

Accidental:

There is no doubt that maturity leveling is accidental. It is basically the use of more advanced and complicated tools and techniques in a team or company. CMM does not define any essential problem. It follows techniques like management controlled software development, self-evaluating software development, etc. Therefore, it is focused on accidental problems.

[1] M. Behrmann, Mike, and Segue's Executive Management, “Why is CMMI Appraisal Important for Development Companies?,” *Segue Technologies*, 13-Oct-2016. [Online]. Available: https://www.seguetech.com/cmmi-appraisal-important-development/. [Accessed: 17-Nov-2019].

Self Evaluation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Exceptional 100% | Good 90% | Acceptable 70% | Developing 50% | Missing 0% |
| Accuracy 60% | The problem all 8 methodologies were meant to address was accurately identified, clearly described, and well supported with direct quotes. | A problem the methodology was accurately identified in all 8 cases. | The problem was not clearly described, the description was vague, or the citation was missing for at least one methodology. | The problem was not clearly described, the description was vague, or the citation was missing for two or more methodologies. | More than one major error exists. |
| Clarity of reasoning 30% | Each Accidental / Essential argument was convincingly made. | There is no flaws in the reasoning for any of the 8 accidental / essential arguments. | The reasoning for one model was incomplete, vague, or otherwise flawed. | A single large flaw exists in the reasoning or presentation of facts for any of the models. | More than one large flaw exists. |
| Writing 10% | The paper is a joy to read and easy to grade. | The paper is professionally written, lacking any mechanical or formatting errors. | One aspect of the paper is difficult to read or there exists a single mechanical error. | Flaws in the paper make it difficult to read or grade. | The paper can be read only with extreme difficulty. |