

Assignment 1

1) Waterfall

The Waterfall process model gets its name from the way the process flows. Work is done in discrete stages that move in one direction. This is similar to a waterfall flowing from up high at the first stage of development down to the last, and never in the other direction. By discrete, I mean that all the work of a particular stage must be completed and verified before the next stage can begin. If an issue arises, Waterfall projects reset to the appropriate earlier stage and follow this same process. The stages are: 1) Requirements Analysis, where the needs of the software are determined, 2) System Design, where the means of meeting these requirements are determined, both in terms of software and hardware, 3) Implementation, in which the software is built, 4) Verification, where the software is tested, and 5) Maintenance, where changes and upkeep are handled.

Advantages

- No overlapping activities due to discrete stages
- Easy to manage because of discrete stages and linear flow
- Simple to learn because it is rigid in design
- Resistant to change because requirements are established up front

Disadvantages

- Rigid as a downside because it is difficult to correct errors in earlier stages and cannot adapt to change easily
- No working code for a long time
- No testing till even later in the life cycle than working code is achieved

2) Spiral

The Spiral process develops its spiral shape by performing iterations of its 4 phases. The first phase involves determining the requirements for the software. In the second phase, risks and alternate solutions are identified. Here a prototype is developed. The third phase is where development and testing is done. The last phase involves evaluating the current cycle and state of development, as well as planning for the next cycle.

Advantages

- The iterative process helps react to new opportunities and risks, as well as changes in customer wants
- An iteration is much shorter than any entire process life cycle, meaning tangible results (software) can be obtained much more quickly than in, for example, the Waterfall model
- As with the Waterfall model, there are distinct phases

Disadvantages

- Intimidating to learn because of the repeating but varied nature of each cycle with challenges unique to each project
- Despite having 4 clear phases, managing a Spiral process is difficult in part because of the complexities of creating effective iterations
- Risk analysis being iterated over each cycle requires expertise that makes the process more difficult to execute well

3) RUP

The Rational Unified Process model can be depicted in tabular form by considering the phases as columns and the types of work being done as rows. It is a highly organized and complex model that leverages tools such as UML and use case diagrams to serve as a foundation for the process. The first phase, inception, defines the scope and requirements of a project. Here a very simple prototype may be created to gauge the project's feasibility. Next, elaboration consists largely of planning and design work. Here the architecture of the project is established. In the third phase, construction, the software is built and brought to a mostly complete product. Design work tapers off as the system comes together, and testing ramps up as components begin to pile up. In the transition phase, the project testing represents the bulk of the workflow, with implementation reacting to needs uncovered through testing and deployment.

Advantages

- Highly organized
- Excellent documentation creates a strong foundation each workflow can benefit from in each phase
- Extensive modeling creates a language that can be shared across disciplines due to the high level of abstraction

Disadvantages

- Very complex, requiring highly skilled teams
- Projects with rapidly changing characteristics, requirements, or risks benefit less from the large amount of documentation that normally feeds phases continually as development moves forward
- Small, fast-moving projects aren't very compatible with RUP's life cycle

4) Scrum

Scrum is a model based around quick development cycles producing constant results. With cycles called sprints lasting less than a month, developers are always reacting to the latest changes in risks, customer needs, and changes in priorities. Work is divided into User Stories, which each represent a type of user, what they want, and the reason they want it. Three key groups of concepts define scrum: artifacts, roles, and ceremonies.

Three artifacts reflect work management. The product backlog consists of user stories that don't have an immediate plan for implementation but is important to the project. The sprint backlog holds all of the user stories being tackled in the current development cycle. The burndown chart is used to track work done during a sprint, showing who contributes to which user stories and when. This is a performance measure used continuously to improve a team's effectiveness.

Three roles define a scrum project. The product owner determines the priorities of the project and is responsible for its business aspects. The scrum master is like a project manager, managing the project and team. They do not take part in development. Instead, they work with the development team to ensure individual success and organization of events (known as ceremonies). The development team makes up the third role, and this group develops the product.

Three events, or ceremonies, are key to scrum. Sprint planning takes place before each sprint. It is a brief activity meant to guide the coming cycle by evaluating the difficulty of the relevant user stories and covering broad issues that may arise. Daily scrum meetings are a way for the scrum master to interface with the development team briefly every day to what they are working on, what they have achieved, and whether they are experiencing any issues. A sprint review meeting occurs at the conclusion of every sprint to evaluate its successes and failures.

SER416-20546

Ryan Schumacher (rschuma2)

Advantages

- Focus on doing over planning
- Lightweight with some management aspects
- Quick iteration very good at adapting to changes

Disadvantages

- Difficult to effectively use meeting / review time without cutting into productivity
- Little created in the way of documentation leads to lower amounts of reuse of work
- Changes in development team composition can be devastating due to how lean the framework is
- Easy to participate in the process without taking advantage of its benefits.
Simple issues like not utilizing Burndown charts and ineffectively generating user stories can go unnoticed and reduce the value of using the model significantly