SW Development Process Guideline

Prepared for Software Engineer

Prepared by: Dan Nguyen, SW Director

Update on April 13, 2020. First version is from Aug, 2016.

- Re-structure document section
- Adding best practice section

EXECUTIVE SUMMARY

Software Engineering deals with the software product as an engineering effort. Software engineering has two main components:

- Techniques: Programming languages, Programming paradigm, Databases, Graphical user interfaces, Command line interfaces, Programming tools, Libraries, Design languages, Patterns, Platforms, Other Practices, Other tools..etc.
- Processes and methodologies: Agile, Waterfall, Scrum, Kanban..etc.

The aim of this document is to provide guidance to software designers and developers by defining a set of criteria for Agile software development methodologies. The criteria established are based on various internationally recognized standards and best practices and some of the processes developed by many key role players.

AGILE PROCESS AND METHODOLOGIES

In general, there are 2 main types of software development methodologies Waterfall model and Agile model.

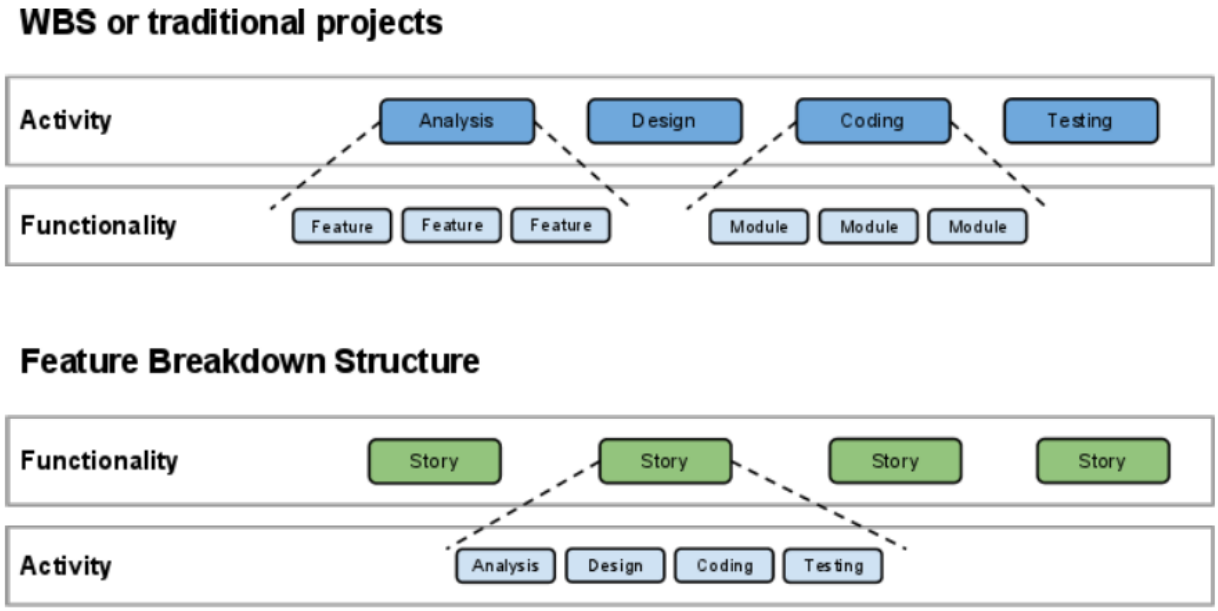
What is the Waterfall model?

A waterfall model represents a linear and sequential approach to software development. The following steps are carried out sequentially in a waterfall approach.

- **Requirements**: gather, analyze, document
- **Design**: resulting in the software architecture
- **Code**: the development, proving, and integration of software
- **Testing**: the systematic discovery and debugging of defects
- **Operations**: the installation, support, and maintenance of complete systems

In an actual Waterfall implementation, each of the above steps represents a different stage of software development, and every stage generally terminates before the next stage begins. There exists a stage-gate between each; e.g., before the designing stage can commence, it's necessary to have customer's approval.

What is the Agile model?



Define the project plan in terms of what will be delivered rather than what work steps will be performed.

Agile methodology focuses on flexibility, continuous improvement, and speed. The Agile methodology follows an incremental approach.
Developers start with simplistic project design and then begin to work on small modules. The work on these modules is carried out in weekly or monthly sprints, and at the end of each sprint, project priorities are evaluated, and tests are executed. Thus, the equivalent stages of the waterfall model are completed at every single sprint. These sprints allow for exploring bugs, and customer feedback to be incorporated into the design before the next sprint begins.
That said, the process must be well-managed and documentation enforced, as the rapid pace of development can lend itself towards dis-organization. However, if done correctly, it results in a fast, low-risk, and efficient workflow. Most of the matured companies now use Agile methodology.

Benefit of the Agile software development methods are:

As examples, releasing working software early and throughout the development process allows users and domain experts to give continuous, valuable, and relevant feedback. This

encourages the exchange of new ideas, a cooperative and interactive software development environment, and should result in the creation of valuable and useful software. Agile software development is a people-oriented, collaborative development approach with the flexibility to respond quickly to changes. The Agile methodology family includes techniques such as Extreme Programming (XP), SCRUM (project management framework), LEAN, Feature-Driven development (FDD), Test-Driven development (TDD), and Dynamic Systems Development Method (DSDM). There are many options and no specific engineering practices are prescribed.

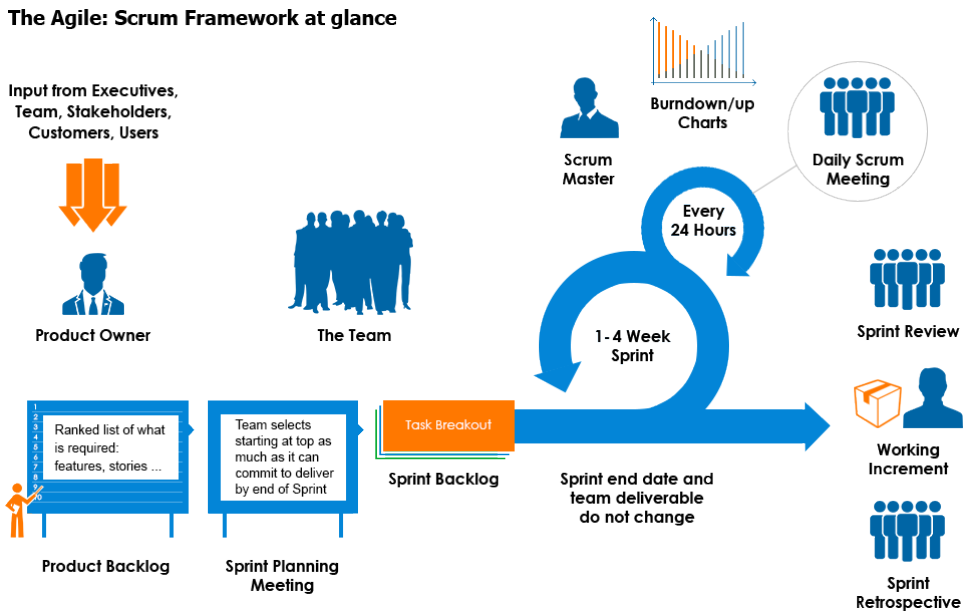
Principles behind the Agile Manifesto:

We follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

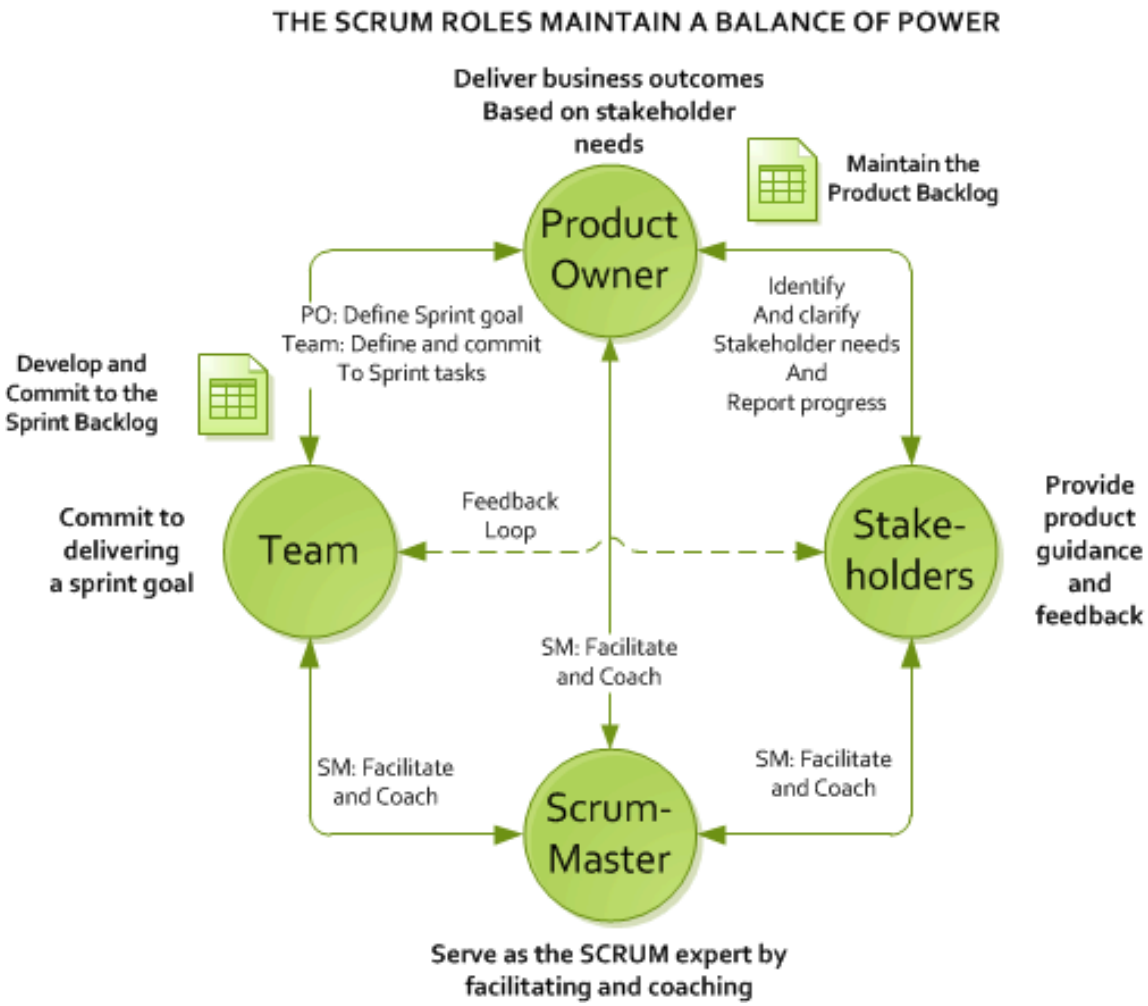
APPLY SCRUM FOR SW DEVELOPMENT

SCRUM is a very specific framework of agile that focuses on the following roles:



- **ScrumMaster:** is a facilitator, coach and champion of the Scrum framework and process. Is not the Agile term for a project manager. Does not have responsibility for delivering the project.
- **Product Owner:** Is responsible for delivering product value. This means working with the Stakeholders to understand their needs and developing a Vision and a Product Backlog that aims to achieve the Vision. Does not control what the Team commits to completing. The Product Owner prioritizes the list of items on the Product Backlog in sequential order (from 1 to n). But the Product Owner cannot force the Team to commit to delivering beyond what the Team feels comfortable.
- **Team:** Is ultimately responsible for committing to a Sprint goal and promising to deliver it within the time boxed Sprint. Is self-managing and self-organized. Sprints cannot be extended or delayed.
- **Stakeholders:** Is anyone who has an interest or stake in the project. This can be the direct managers of the Team members, the persons providing funding for the project, the Product Owner, a Team member.

SCRUM Roles responsibilities



ScrumMaster responsibilities include the following:

- Communicate the value of Scrum
- Teach the organization on Scrum to maximize business value
- Facilitate Sprint Planning, Daily Scrums, Sprint Reviews and Retrospective Meetings
- Create the Task Board and Sprint Burndown Chart at the start of every Sprint
- Attend all Scrum meetings
- Preserve the integrity and spirit of the Scrum framework

- Maintain the focus of the Team
- Make the Team aware of impediments and facilitate efforts to resolve them
- Serve as a coach and mentor to members of the Team
- Respectfully hold the Team, Product Owner and Stakeholders accountable for their commitments
- Continually work with the Team and business to find and implement improvements

Product Owner responsibilities include the following:

- Maximize business value by the Team
- Maintain and prioritize the Product Backlog sequentially (1 to n)
- Create and maintain the Release Burn down Chart
- Help the ScrumMaster organize Sprint Review Meetings
- Attend Scrum Meetings
- Clearly communicate the business case to the Team and Stakeholders
- Build and maintain a relationship with the Stakeholders
- Support the ScrumMaster to help the Team become self-organizing
- Report progress to the Stakeholders regularly
- Ensure the proper use of corporate resources and assist the Team to obtain resources as needed

Team responsibilities include the following:

- Commit to, and self-organize, around a Sprint Goal (A Sprint Goal is different from the Sprint activities. The Sprint Goal is the intended spirit and purpose of the Sprint. So if the Team realizes in mid Sprint that a required activity for the Sprint is missing, the Team should add the activity to the Sprint so that they can deliver the Sprint Goal)
- Work with the Product Owner to analyze and decompose the Product Backlog items
- Help create and maintain the Sprint Backlog, Sprint Burn down Chart and Task Board

- Demonstrate the product at the end of each Sprint — during the Sprint Review
- Implement action items that come out of Retrospectives (essentially lessons learned)
- Facilitate Sprint Planning, Daily Scrums and Retrospectives if the ScrumMaster is not able to do so for any reason
- Attend all Scrum meetings
- Collaborate and share knowledge and experience among the Team, Product Owner, ScrumMaster and Stakeholders
- Help other members of the Team
- Look for ways to continually improve

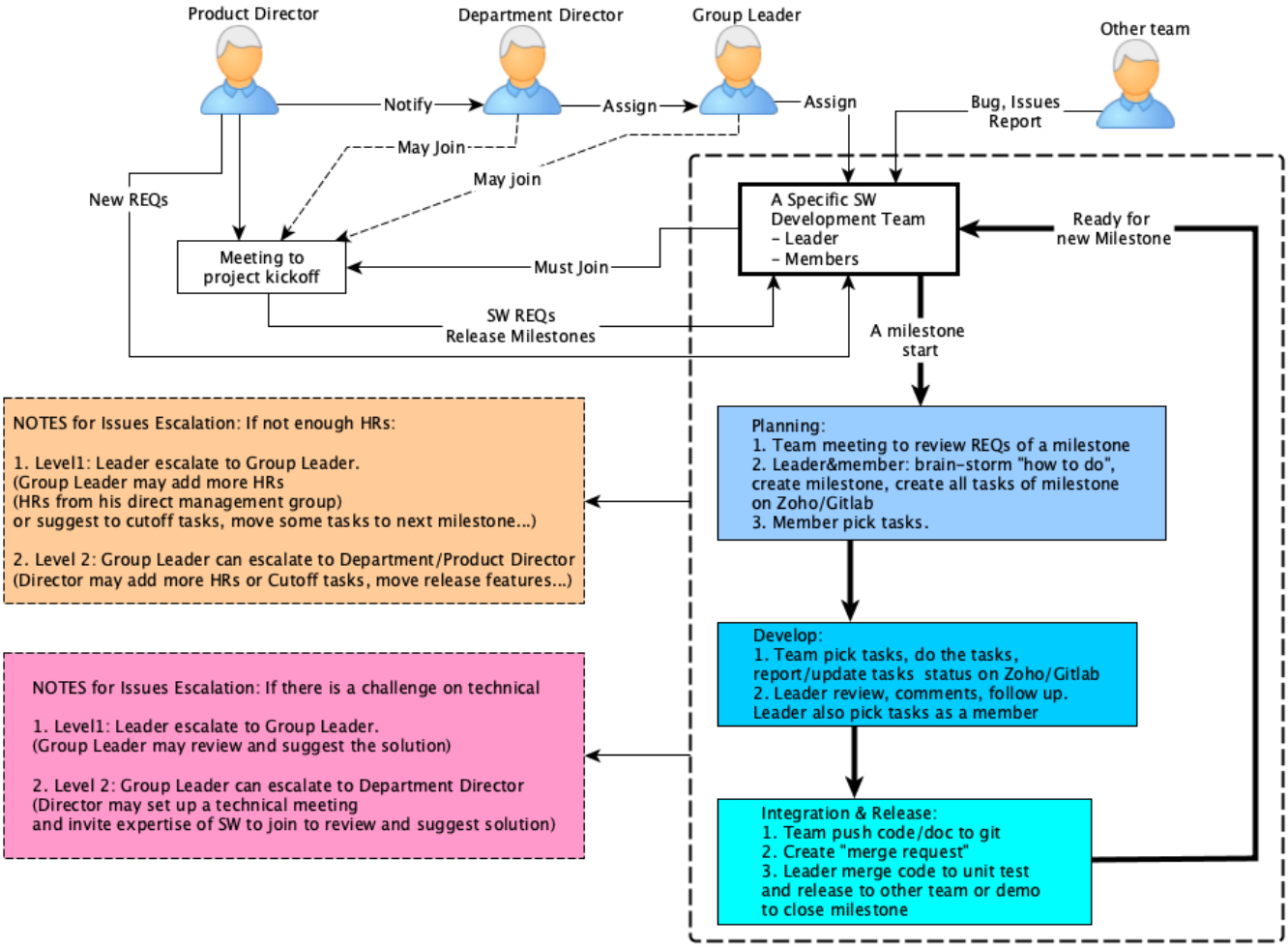
Stakeholder responsibilities include:

- Work with the Product Owner to develop and maintain the Product Backlog
- Attend Sprint Planning meetings as needed to provide feedback and expertise
- Provide direct feedback to the Team during Sprint Reviews
- Remove roadblocks and impediments for the Team, Product Owner and ScrumMaster
- Avoid distracting the Team during a Sprint — after the Team has committed to the Sprint
- Support the Scrum Framework

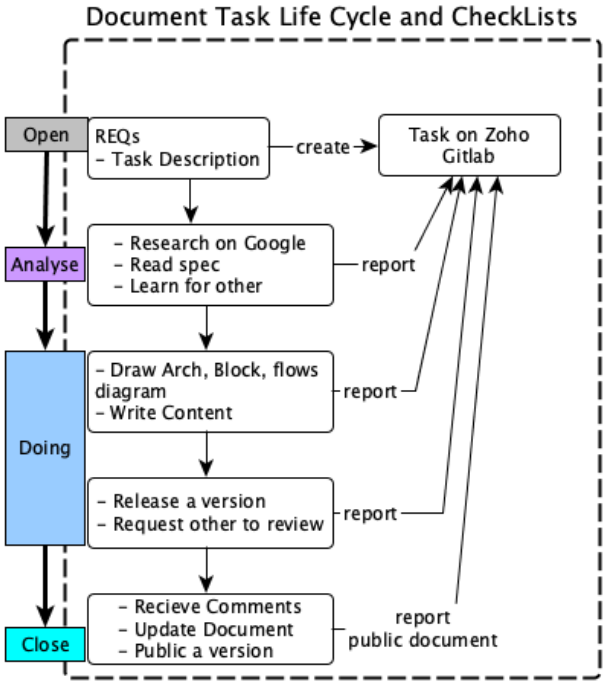
Mapping SW roles into SCRUM

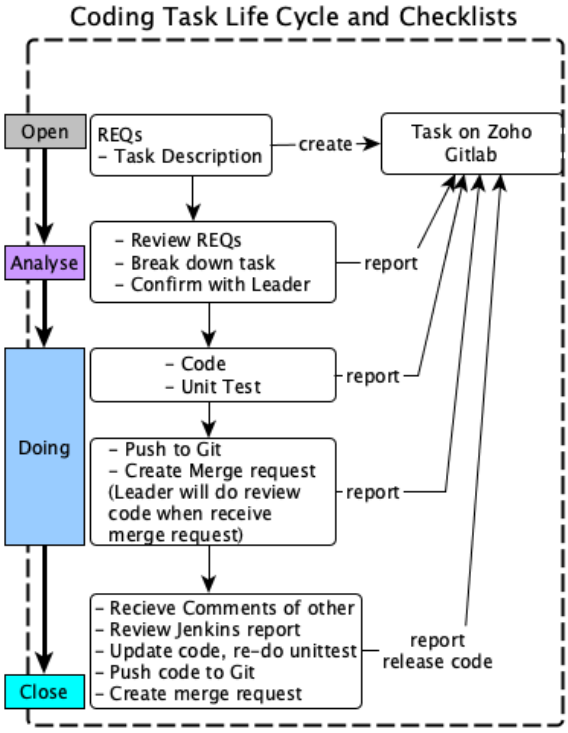
- **ScrumMaster** = Google + SW Directors + SDK Group leader.
- **Product Owner** = Product Line Director + Project Lead + SDK leader
- **Team** = a dedicate group of SW + SDK leader
- **Stakeholder** = **ScrumMaster** + **Product Owner** + **Team** + VP + CS + Customers + ..etc.

PROCESS BEST PRACTICES



TASK LIFE CYCLE





TEMPLATES

General SW Task List:
1. [doc] Write Technical Notes/Presentation/Proposal for a SDK, SW or a small SW module/features
2. [code] Branch/Create SDK/SW code structure for SDK or SW app on Git Repos
3. [code] Prototype Define, Develop, Unittest for a SDK module, SW module
4. [code] Merge/Integrate Code for SDK, SW source release
5. [supp] Do Document review
6. [supp] Do Code Review
7. [supp] Do online/Onisite Support
8. [code] Issues/Bug Analysis and Fixing
9. [supp] Prepare and do demo

General Task Description should have:
- Standard/Spec referrence
- What to do
- When to delivery

General Task Report should have:
- What have been done (Did/Doing)
- What are going to do (Next steps)
- Is there any risk/issue that may affect delivery date?

Technical Notes Template:
Section 1: Introduction
Section 2: Usages/Use Cases/How to Integrate with Other
Section 3: Arch/Diagram/Flows chart
Section 4: Data Type, Data Structure, APIs/Function Prototype
Section 5: How To Test, Test REQs