Software Development in Practice

How software engineering teams really work?

William Chan
Lead Platform Engineer, 605.tv, Capital One, FreeWheel (Comcast)

Course Objective

- Learn real life software engineering processes outside of the classroom
- Practice real life software engineering practices
- Become software engineers!

Course Breakdown

1. Software Development Methodologies
2. Product Development and Requirements
3. Software Design
4. Software Implementation

3

Project Breakdown + Technical Requirements

Choosing a project that will help you succeed in this course for grades sake

- Use a relational database or a non-relational database if you can prove a need for it
- A web or desktop application that stores and reads data from a database
- A working demo by the end of the semester that will be presentable to potential employers

4

Presentation Breakdown

1. **Product**: Discuss the overall product mission and what your application intends to solve; may include business idea and notes
2. **Planning**: Milestones for the application from the beginning of the semester to the end of the semester
3. **Requirements**: List the requirements and reasons behind why that constitutes your MVP
4. **Technical Design**: UML diagrams and additional slides that demonstrates the reason to the design
5. **Tests**: What tests will you have in place to ensure the quality of your application
6. **Demo**: Be able to present a partial or preferably a full demo; if its a partial demo, explain why the project could not be completed as is
7. **Retrospective**: knowing what you know after having implemented the project, what would you have changed or think you can do better?
8. Questions: The ability to defend any section of the presentation

Note: there will be assignments that will be the milestones that can be laid out in the presentation

Work Environment

- Project to resemble a real work environment
- All of you will be broken into teams that work on different projects
- All of you will define the product, gather requirements, design the application, implement the application, test the application and demo the application
- All of you will carry out 360 reviewing each other to make sure that all of you carry the weights of the team

Typical Project Roles

- Product Manager/Owner - someone who is close to the users and knows the problem that the application solves; the stakeholder in the most general sense
- Software Engineers - responsible for technical design and implementation of the product
- Project Managers/SCRUM Master - person responsible for removing obstacles that block software engineers from delivering the product
- Program Manager - typically in larger organizations to manage teams that break down into multiple products
- QA Engineering/Software Engineering in Test - slowly fading away in favor of software engineers being accountable for their own work

What does a generic software development process look like?

1. Define the product
2. Gather requirements
3. Design the application
4. Implement the application
5. Test the application
6. Release the application
7. Gather user feedback

8

Software Methodologies

- Waterfall - lots of time spent thinking about the design and make sure all edge cases are covered, aim to complete the implementation in one cycle and then demo to client
- Kanban - no defined timeline but stories or tickets are groomed constantly to ensure that priorities are always on the top of the backlog
- SCRUM, SAFe, LeSS and other variants - like kanban but in the form of sprints to measure that given a same length cycle, determine the velocity at which the team completes tickets and stories
- Lean - based on doing the minimal viable set every time and work to eliminate wasteful work

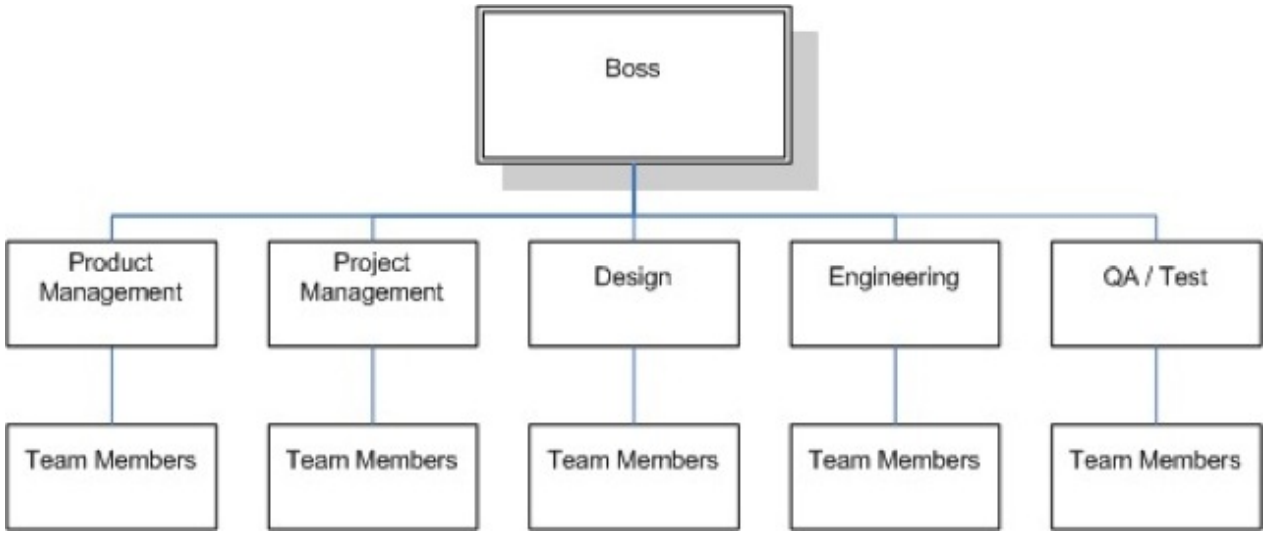
Note: More to come in future slides

9

Software Engineering Organizations

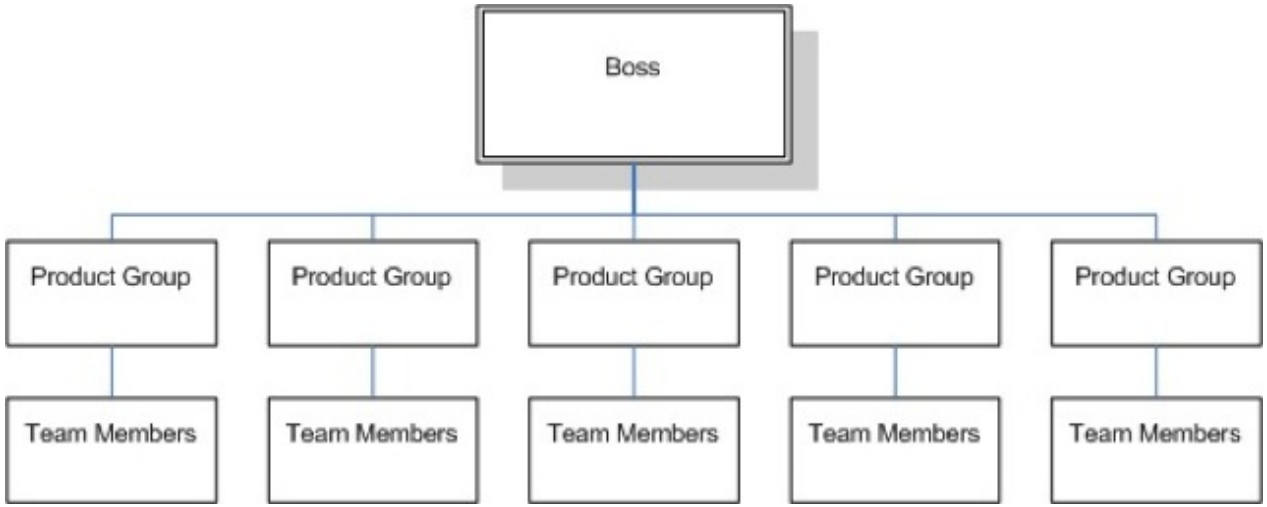
- Functional - groups of people under functions such as front end developers, back end developers, etc
- Divisional - cross functional teams where there may be front end developers, back end developers, product managers, etc
- Matrix - organized around different products that are delivered
- Flat - everyone reports to the CEO or where have minimal layers of managers

The Functional Organization



Source: http://itsadeliverything.com/using-a-product-led-matrix-in-lean-agile

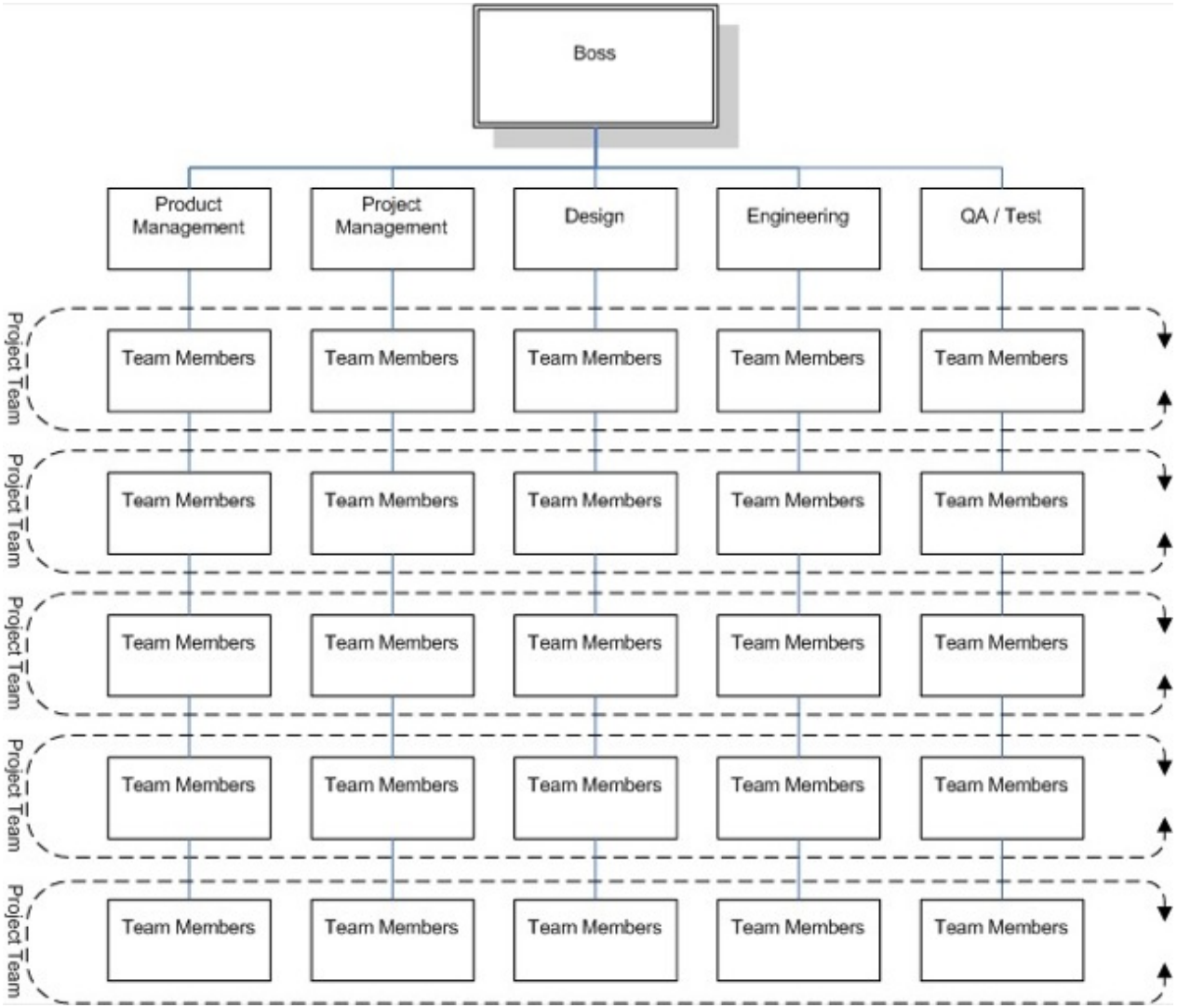
The Divisional Organization



Source: http://itsadeliverything.com/using-a-product-led-matrix-in-lean-agile

The Matrix Organization

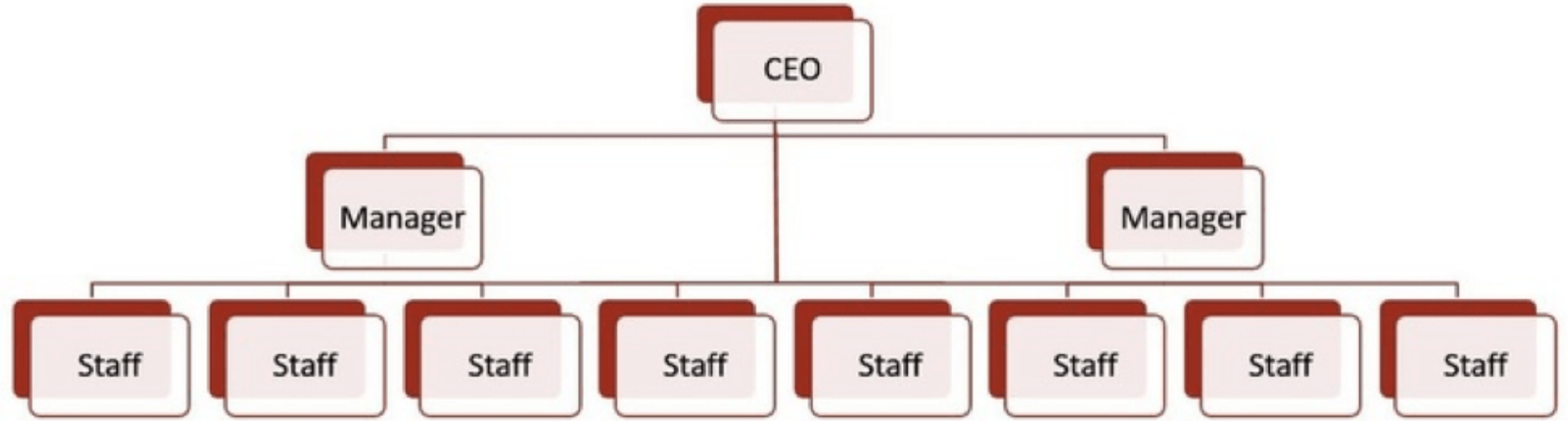
The Matrix Organization



Source: http://itsadeliverything.com/using-a-product-led-matrix-in-lean-agile

13

The Flat Organization



Source: https://pingboard.com/org-charts/evolution-org-charts

14

Software Engineering Career Tracks

- Managerial - people manager, generally responsible for the well being of the team and growing people
- Technical - growing people by technical means and ensure technical delivery of a project

15

Software Engineering Culture

- Distributed teams - remote workers organized in small cross functional teams usually
- Flexible work culture to breed creativity in engineers
- Unlimited vacation days
- Conference budgets

16

Software Development Tools

- Version Control: Git, SVN, CVS, Mercurial, Bazaar

- Version Control: Git, SVN, CVS, Mercurial, Bazaar
- Repository Management: GitHub, GitLab, Bitbucket
- Project Management: Jira, Pivotal Tracker, Trello, Asana
- UML Diagrams: PlantUML, yUML, Mermaid
- Text Editors/IDE: Intellij IDEA, Eclipse, Sublime Text 2/3, Atom, Visual Studio Code

Product Management & Project Management Workflow

1. Collaborate with software engineers to decide what the breakdown of the tasks are

2. Define the **acceptance criteria** for which a product will accept a feature as complete
3. Software engineers estimate the complexity or the time of each story
4. Product managers and project managers define the priorities with the estimates in mind

18

Software Development Workflow

1. Technical design - technical approach to solving the problems; high level components and

how they interact with each other
2. Implementation - code is written and logic is verified
3. Open a pull request - became popular through the adoption of GitHub with distributed version control
4. Code review - ensures standards and quality
5. Merge to a branch to be deployed in a QA or staging environment that mimics production
6. Feature validated by the product owner and get pushed to production through an approval process

Summary

- Software engineering is an iterative and team based process

- There are often two career tracks, the managerial and technical tracks in engineering organizations
- Version control, project management tools, UML diagrams, text editors, and repository management is often used to help software engineering teams be productive and collaborate

20

Thank you

William Chan
Lead Platform Engineer, 605.tv, Capital One, FreeWheel (Comcast)

wchan@605.tv (mailto:wchan@605.tv)

http://linkedin.com/in/wchan2 (http://linkedin.com/in/wchan2)