Agile Process

for software development team management

- What is agile software development?
- What advantages does it bring?
- Where is it appropriate?

The “Agile Process” is a strategy for software development team management. I will discuss what it is, what its advantages are, and what situations call for it.

Elliot Chandler, 2018 May 15a16, for NMD306

What is agile software development?

- Project management strategy
- Develop useful software despite frequently changing requirements
- Fewer known goals and documentation
- “Iterations” of work

- A strategy for project management in software development teams working for corporate clients
- Intended to let a team develop useful software without knowing exactly what it is supposed to do at the start of the project, adapting to frequently changing requirements (hence the term “agile”)
- To adapt quickly, avoids predefined end goals or detailed documentation
- Work happens usually in one to four week “iterations”, also called “sprints”, of a predetermined duration, after which a working product is demonstrated to the client
- Most planning is done on a per-iteration basis by the team members, rather than by management

What advantages does agile software development bring?

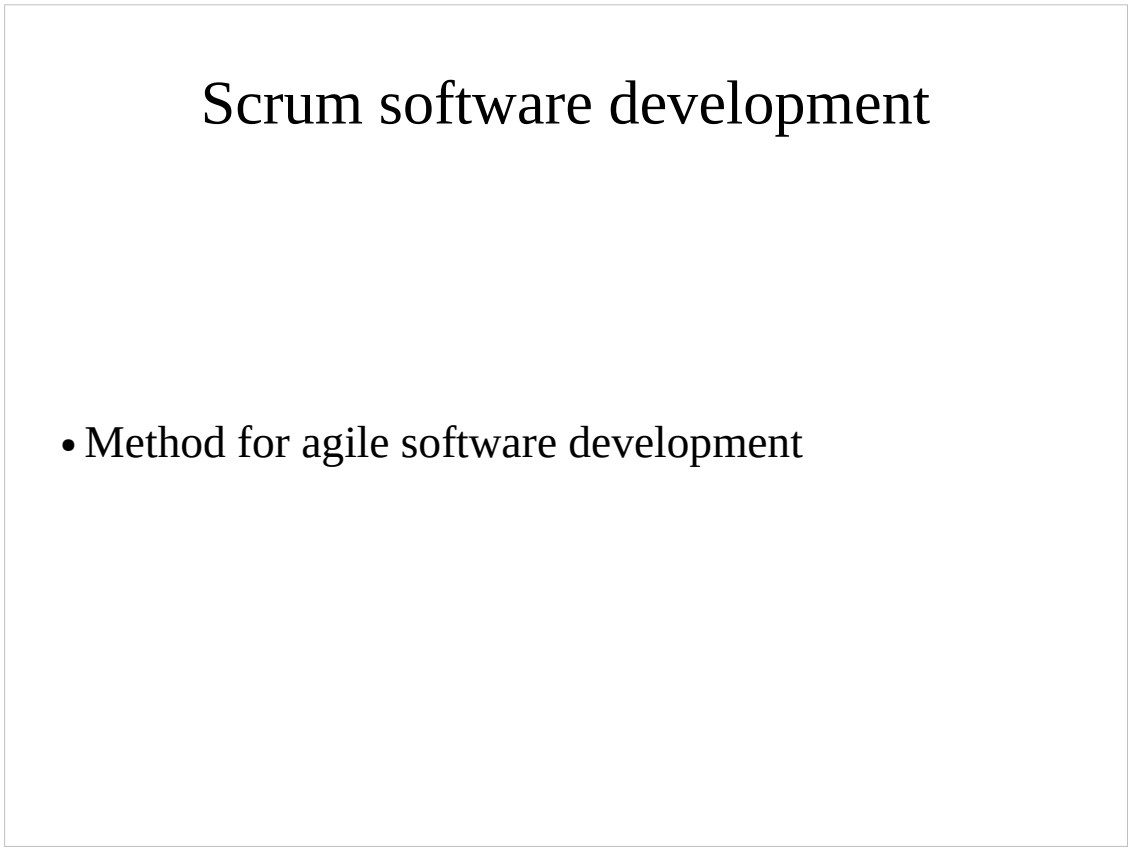
- Work collaboratively with clients
- Avoid expensive losses for clients

- Promotes collaboration with corporate clients. In the traditional software development process, a client will provide a specification for the software, and that will be implemented. The resulting implementation, when actually used, often reveals deficiencies in the original specification, which is expensive for the clients due to requiring a rewrite or scrapping the project.
- By frequent iterations each delivering usable software and receiving feedback, and not basing projects around a specification agreed upon at the beginning of the project, agile development attempts to mitigate these losses for the clients.
-

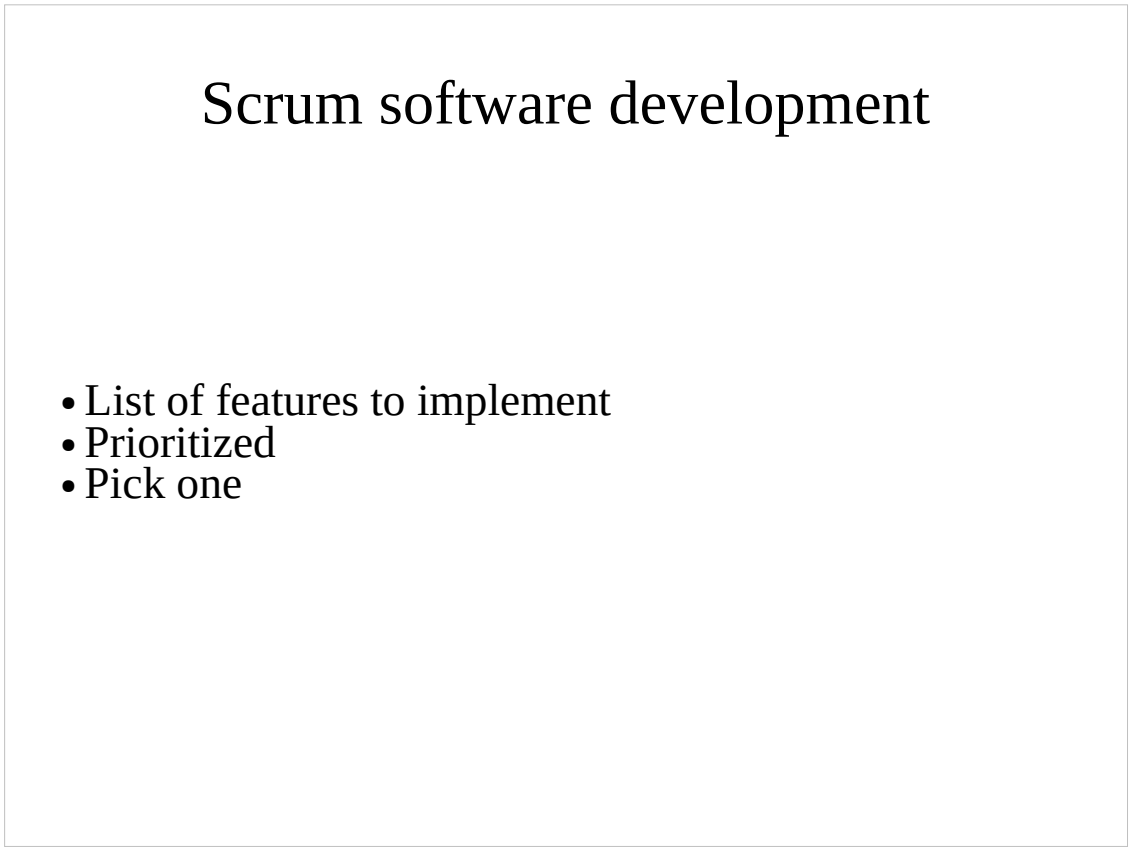
Where is agile software development appropriate?

- When various different end products could be good
- When goals are unknown or changing
- But less when goals are exactly known

- Agile software development is useful when the end goals of a project are not known, or have a possibility of changing.
- Conversely, it is less useful when the end goals are exactly known and guaranteed not to change.
- Thus, it is generally more useful for new projects, especially when there is a range of acceptable end products, than for projects with fixed goals (such as reimplementing an existing system while maintaining API compatibility).



- Scrum software development is a structured method for pursuing agile software development.



- In scrum software development, first, the person in charge of the project defines a prioritized list of features to implement.

Scrum software development

- Series of iterations
  - Feature to implement chosen
  - Daily meeting: discuss challenges and progress
  - Review

- A series of iterations is then undertaken.
- Within each iteration,
  - first, a feature to implement during the iteration is chosen.
  - Then, each day during the iteration, the team meets and discusses the challenges and progress they have encountered; this meeting happens under the guidance of a designated person.
  - Finally, the results of the work and the team's efficiency during the sprint are reviewed as a group.
- The iterations will continue until (quote) "the deadline has been reached, the budget is exhausted, or the product owner is satisfied with the final product" (end quote).

Two questions from the reading

- Are there elements of agile software development that can be useful for individuals' projects? What about community-maintained free/libre/open-source projects?
- "Customer collaboration" (quoted in Schneider p. 12) is a theme of agile software development, but the implementation of it seems risky. When a product specification is only provided informally during development, how can a company protect themselves from a client accusing the company of not having fulfilled their requests effectively? Conversely, how can a client protect themselves from a company that implements the software carelessly? A contracted up-front specification has the advantage of heading subjective battles off at the pass, since correctness of the software can be objectively determined.

Sources

- Johnny Schneider. *Understanding Design Thinking, Lean, and Agile*. O'Reilly Media, 2007.
- "Understanding Agile — In Simple Words" http://www.abodeqa.com/understanding-agile-in-simple-words/
- "A Beginners Guide To Understanding The Agile Method" https://linchpinseo.com/the-agile-method/
- "Agile & Waterfall Methodologies – A Side-By-Side Comparison" http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/
- "Agile Development — Advantages, Disadvantages, and when to use it?" https://www.360logica.com/blog/agile-development-advantages-disadvantages-and-when-to-use-it/
- "Three Myths About Agile in Regulated Environments" https://www.ca.com/en/blog-highlight/agile-audits-apples-oranges-three-myths-agile-regulated-environments.html
- "What is Scrum Methodology? A Beginner's Primer" https://blog.capterra.com/what-is-scrum/
- "The Beginner's Guide To Scrum And Agile Project Management" https://blog.trello.com/beginners-guide-scrum-and-agile-project-management
- "Scrum Explained — In Simple, Plain English without Jargon" https://www.linkedin.com/pulse/scrum-explained-simple-plain-english-ravulapalli