Steve McConnell

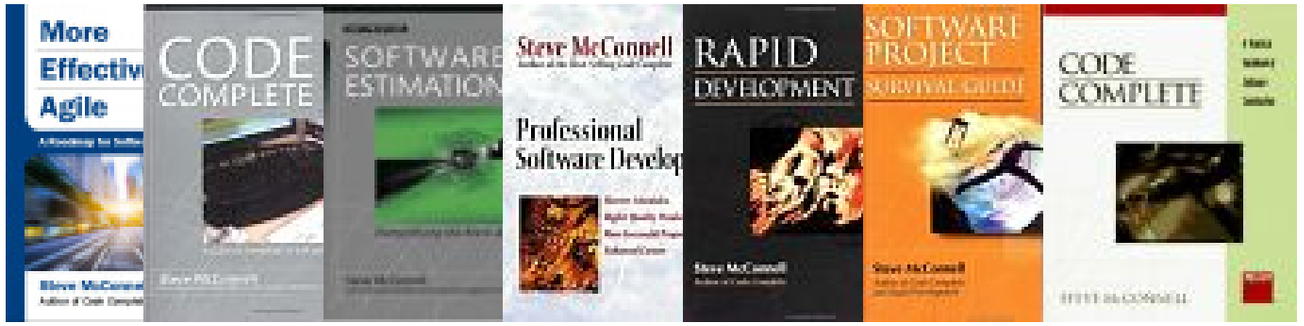
CSCI 4448/5448: Object-Oriented Analysis & Design

Lecture 21a

Who?

- Steve McConnell
- Well known author of software engineering textbooks
- Worked at Microsoft, Boeing, and other Seattle area businesses
- Has a training business now, Construx
- https://stevemcconnell.com/

What books?



- **More Effective Agile: A Roadmap for Software Leaders** - **M**odern Agile practices that work best. Check out the resources on the More Effective Agile website.
- **Code Complete 2.** (Now also an online course) A practical handbook of software-construction practices. Updated for web development, object-oriented development, agile practices, and other modern construction issues.
- **Software Estimation: Demystifying the Black Art**. A comprehensive set of tips and heuristics that software developers, technical leads, and project managers can apply to create more accurate estimates
- **Professional Software Development**. Essays about the software engineering profession

Older books:

- **Rapid Development**. Strategy and best practices for optimizing software development schedules. This book might still have some historical interest but is pretty out of date at this point.
- **Software Project Survival Guide**. A step-by-step guide to running a successful software project. This was a good book in its day, but today I would recommend an introductory Scrum book instead.
- **Code Complete, 1st Edition**. A practical handbook of software-construction practices. Superseded by Code Complete 2. This is still available as a new book on Amazon (inexplicably, since it's been out of print for 15 years), but the second edition is strongly preferred at this point.
- **After the Gold Rush** (out of print). Essays about creating a true profession of software engineering. This book has been superceded by Professional Software Development

Pearls of Experience and Wisdom

- Rapid Development – Rescuing a Project
- Classic Software Mistakes
- Customer and Project Team Bill of Rights
- Software Project Survival Test

Classic Software Mistakes

- Undermined motivation.
- Uncontrolled problem employees.
- Noisy, crowded offices.
- Abandoning planning under pressure.
- Shortchanging upstream activities.
- Shortchanging quality assurance to improve development speed.
- Lack of feature-creep control.
- Silver-bullet syndrome.
- Wasting time in the “fuzzy front end.”
- Insufficient user input.
- Overly aggressive schedules.
- Adding developers to a late project.
  - Fred Brooks likened adding people to a late project to pouring gasoline on a fire (*Mythical Man-Month*, Addison Wesley, 1975).
- https://stevemcconnell.com/articles/classic-mistakes/

Customer's Bill of Rights

I have the right to…

- To set objectives for the project and have them followed
- To know how long the software project will take and how much it will cost
- To decide which features are in and which are out of the software
- To make reasonable changes to requirements throughout the course of the project and to know the costs of making those changes
- To know the project's status clearly and confidently
- To be apprised regularly of risks that could affect cost, schedule, or quality, and to be provided with options for addressing potential problems
- To have ready access to project deliverables throughout the project

Project Team's Bill of Rights

We have the rights to…

- To know the project objectives and to clarify priorities.
- To know in detail what product I'm supposed to build and to clarify the product definition if it is unclear.
- To have ready access to the customer, manager, marketer, or other person responsible for making decisions about the software's functionality.
- To work each phase of the project in a technically responsible way, especially to not be forced to start coding too early in the project.
- To approve effort and schedule estimates for any work that I will be asked to perform. This includes the right to provide only the kinds of cost and schedule estimates that are theoretically possible at each stage of the project; to take the time needed to create meaningful estimates; and to revise estimates whenever the project's requirements change.
- To have my project's status reported accurately to customers and upper management.
- To work in a productive environment free from frequent interruptions and distractions, especially during critical parts of the project.

Software Project Survival Test

- Waterfall-ish project health check
- Looks at Requirements, Project Control, Risk Management, Personnel
- http://users.csc.calpoly.edu/~jdalbey/206/Assign/survivetest.html

Software Project Survival Test

Requirements

- Does the project have a clear, unambiguous vision statement or mission statement?
- Do all team members believe the vision is realistic?
- Does the project have a business case that details the business benefit and how the benefit will be measured?
- Does the project have a user interface prototype that realistically and vividly demonstrates the functionality that the actual system will have?
- Does the project have a detailed, written specification of what the software is supposed to do?
- Did the project team interview people who will actually use the software (end users) early in the project and continue to involve them throughout the project?
- Does the project have a detailed, written Software Development Plan?
- Does the project's task list include creation of an installation program, conversion of data from previous versions of the system, integration with third-party software, meetings with the customer, and other "minor" tasks?

Software Project Survival Test

Requirements Continued

- Were the schedule and budget estimates officially updated at the end of the most recently completed phase?
- Does the project have detailed, written architecture and design documents?
- Does the project have a detailed, written Quality Assurance Plan that requires design and code reviews in addition to system testing?
- Does the project have a detailed Staged Delivery Plan for the software, which describes the stages in which the software will be implemented and delivered?
- Does the project's plan include time for holidays, vacation days, sick days, and ongoing training, and are resources allocated at less than 100 percent?
- Was the project plan, including the schedule, approved by the development team, the quality assurance team, and the technical writing team, in other words, the people responsible for doing the work?

Software Project Survival Test

Project Issues

- Has a single key executive who has decision-making authority been made responsible for the project, and does the project have that person's active support?
- Does the project manager's workload allow him or her to devote an adequate amount of time to the project?
- Does the project have well-defined, detailed milestones ("binary milestones") that are considered to be either 100 percent done or 100 percent not done?
- Can a project stakeholder easily find out which of these binary milestones have been completed?
- Does the project have a feedback channel by which project members can anonymously report problems to their own managers and upper managers?
- Does the project have a written plan for controlling changes to the software's specification?
- Does the project have a Change Control Board that has final authority to accept or reject proposed changes?
- Are planning materials and status information for the project, including effort and schedule estimates, task assignments, and progress compared to the plan thus far, available to every team member?
- Is all source code placed under automated revision control?
- Does the project environment include the basic tools needed to complete the project, including defect tracking software, source code control, and project management software?

Software Project Survival Test

Risk Management

- Does the project plan articulate a list of current risks to the project? Has the list been updated recently?
- Does the project have a project risk officer who is responsible for identifying emerging risks to the project?
- If the project uses subcontractors, does it have a plan for managing each subcontract organization and a single person in charge of each one? (Give the project full score if it doesn't use subcontractors.)

Personnel

- Does the project team have all the technical expertise needed to complete the project?
- Does the project team have expertise with the business environment in which the software will operate?
- Does the project have a technical leader capable of leading the project successfully?
- Are there enough people to do all the work required?
- Does everyone work well together?
- Is each person committed to the project?