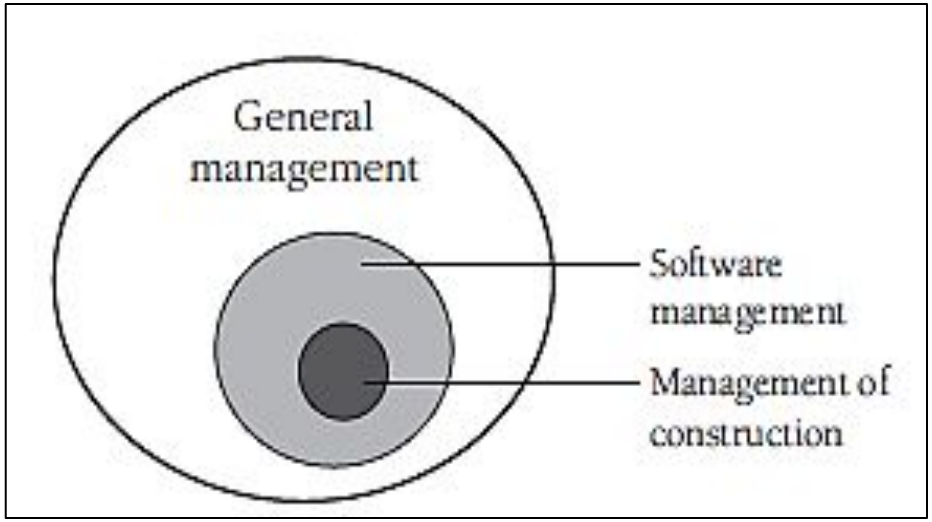
Managing Construction

Contents

- Introduction
- Encouraging Good Coding
- Configuration Management
- Estimating a Construction Schedule
- Measurement
- Key Points

Introduction

- Management is the process of dealing with or controlling things or people.
- Management is the organization and coordination of the activities of a business in order to achieve **defined** objectives.



Introduction

- Factors that significantly affect how a specific software project should be managed are
  - Either sequential or iterative approach is being used
  - Quality goals
  - Size of the project

Encouraging Good Coding

- Some techniques for encouraging good coding are
  - Assign two people to every part of the project
  - Review every line of code
  - Require code sign-offs
  - Route good code examples for review
  - Reward good code
  - One easy standard

Configuration Management

- A software project is dynamic.
- The code changes, the design changes, and the requirements change.
- Configuration management is the practice of identifying project artefacts and handling changes systematically so that a system can maintain its integrity over time.
- Another name for it is ***"change control."***
- It includes techniques for evaluating proposed changes, tracking changes, and keeping copies of the system as it existed at various points in time.

Configuration Management

- If there is no “change control” procedure
  - A code can be written that is incompatible to the other parts of the system.
  - You might change a routine that someone else is changing at the same time; successfully combining your changes with theirs will be problematic.
  - You can make changes to a routine, find new errors, and not be able to back up to the old, working routine.

Requirements and Design Changes

- Some guidelines for controlling design changes are: *(details from book, pg. 666,667)*
  - Follow a systematic change-control procedure
  - Handle change requests in groups
  - Estimate the cost of each change
  - Be wary of high change volumes
  - Establish a change-control board

Software Code Changes

- Another configuration-management issue is controlling source code.
- If you change the code and a new error surfaces that seems unrelated to the change you made, you'll probably want to compare the new version of the code to the old in your search for the source of the error.
- If that doesn't tell you anything, you might want to look at a version that's even older.
- This kind of excursion through history is easy if you have version control tools that keep track of multiple versions of source code.

Software Code Changes

- Good version-control software works so easily that you barely notice you're using it. It's especially helpful on team projects.
- One style of version control locks source files so that only one person can modify a file at a time.
- Another style allows multiple people to work on files simultaneously and handles the issue of merging changes when the code is checked in.
- In either case, when you check the file in, version control asks why you changed it, and you type in a reason.

Software Code Changes

- Some benefits of using a version control software is
  - You don't step on anyone's toes by working on a file while someone else is working on it (or at least you'll know about it if you do).
  - You can easily update your copies of all the project's files to the current versions, usually by issuing a single command.
  - You can backtrack to any version of any file that was ever checked into version control.
  - You can get a list of the changes made to any version of any file.
  - You don't have to worry about personal backups because the version control copy is a safety net.

Backup Plan

- A backup plan is the idea of backing up your work periodically.
- Many things can happen to computerized data: a disk can fail; you or someone else can delete key files accidentally; an angry employee can sabotage your machine; or you could lose a machine to theft, flood, or fire.
- Take steps to safeguard your work.
- Your backup plan should include making backups on a periodic basis and periodic transfer of backups to off-site storage.
- It should encompass all the important materials on your project—documents, graphics, and notes—in addition to source code.

Backup Plan

- When you finish a project, make a project archive.
- Save a copy of everything: source code, compilers, tools, requirements, design, documentation, everything you need to re-create the product.
- Keep it all in a safe place.

Estimating a Construction Schedule

- Managing a software project is one of the formidable challenges of the twenty-first century, and estimating the size of a project and the effort required to complete it is one of the most challenging aspects of software-project management.
- The average large software project is one year late and 100 percent over budget.

Estimating a Construction Schedule

- Some effective size and effort estimation approaches are
  - Use estimating software.
  - Use an algorithmic approach, such as Cocomo II.
  - Have outside estimation experts estimate the project.
  - Have a walk-through meeting for estimates.
  - Estimate pieces of the project, and then add the pieces together.
  - Have people estimate their own tasks, and then add the task estimates together.
  - Refer to experience on previous projects.

Estimating a Construction Schedule

- A good approach for estimating a project should include following activities
- ***Establish objectives:***
  - Why do you need an estimate? What are you estimating?
  - Are you estimating only construction activities, or all of development?
  - Are you estimating only the effort for your project, or your project plus vacations, holidays, training, and other non-project activities?
  - How accurate does the estimate need to be to meet your objectives?

Estimating a Construction Schedule

- ***Allow time for the estimate, and plan it:***
  - Rushed estimates are inaccurate estimates.
  - If you're estimating a large project, treat estimation as a mini project and take the time to miniplan the estimate so that you can do it well.

Estimating a Construction Schedule

- ***Spell out software requirements:***
  - Just as an architect can't estimate how much a "pretty big" house will cost, you can't reliably estimate a "pretty big" software project.
  - It's unreasonable for anyone to expect you to be able to estimate the amount of work required to build something when "something" has not yet been defined.
  - Define requirements or plan a preliminary exploration phase before making an estimate.

Estimating a Construction Schedule

- ***Estimate at a low level of detail:***
- Depending on the objectives you have identified, base the estimate on a detailed examination of project activities.
- In general, the more detailed your examination is, the more accurate your estimate will be.

Estimating a Construction Schedule

- ***Use several different estimation techniques, and compare the results:***
  - The list of estimation approaches at the beginning of the section identified several techniques.
  - They won't all produce the same results, so try several of them.
  - Study the different results from the different approaches.
  - No approach is best in all circumstances, and the differences among them can be illuminating.

Estimating a Construction Schedule

- ***Re estimate periodically:***
  - Factors on a software project change after the initial estimate, so plan to update your estimates periodically.
  - The accuracy of your estimates generally improve as you move toward completing the project.
  - From time to time, compare your actual results to your estimated results, and use that evaluation to refine estimates for the remainder of the project.

Estimation vs. Control

- Estimation is an important part of the planning needed to complete a software project on time.
- Once you have a delivery date and a product specification, the main problem is how to control the expenditure of human and technical resources for an on-time delivery of the product.
- In that sense, the accuracy of the initial estimate is much less important than your subsequent success at controlling resources to meet the schedule.

Estimation vs. Control

- The average project overruns its planned schedule by about 100 percent.
- When you're behind, you can try one or more of these solutions:
  - Expand the team (Might be risky)
  - Reduce the scope of the project
  - Inform customer timely

Measurement

- Software projects can be measured in numerous ways. Here are some solid reasons to measure your process:
  - For any project attribute, it's possible to measure that attribute in a way that's superior not measuring it at all.
  - Be aware of measurement side effects
  - To argue against measurement is to argue that it's better not to know what's really happening on your project

Measurement

- You can measure virtually any aspect of the software-development process.
- Table 28-2 (pg. 678) lists some useful software development measurements.

Size
Total lines of code written
Total comment lines
Total number of classes or routines
Total data declarations
Total blank lines

Defect Tracking
Severity of each defect
Location of each defect (class or routine)
Origin of each defect (requirements, design, construction, test)
Way in which each defect is corrected
Person responsible for each defect
Number of lines affected by each defect correction
Work hours spent correcting each defect
Average time required to find a defect
Average time required to fix a defect
Number of attempts made to correct each defect

Productivity
Work-hours spent on the project
Work-hours spent on each class or routine
Number of times each class or routine changed
Dollars spent on project
Dollars spent per line of code
Dollars spent per defect
Overall Quality
Total number of defects
Number of defects in each class or routine
Average defects per thousand lines of code
Mean time between failures Compiler-detected errors

Key Points

- Good coding practices can be achieved either through enforced standards or through more light-handed approaches.
- Configuration management, when properly applied, makes programmers' jobs easier. This especially includes change control.
- Good software estimation is a significant challenge. Keys to success are using multiple approaches, tightening down your estimates as you work your way into the project, and making use of data to create the estimates.
- Measurement is a key to successful construction management. You can find ways to measure any aspect of a project that are better than not measuring it at all. Accurate measurement is a key to accurate scheduling, to quality control, and to improving your development process.

Readings

- **[Chapter 28]** Code Complete: A Practical Handbook of Software Construction by Steve McConnell, Microsoft Press; 2nd Edition (July 7, 2004). ISBN-10: 0735619670