Comp8081

Management Issues in Software Engineering

Donna Turner

Agenda

- Attendance
- Review (McConnell – Chapters 1 & 2)
- Industry Terminology
- Classic Mistakes (McConnell – Chapter 3)
- Development Fundamentals (McConnell – Chapter 4)
  - Tools and Techniques
- Summary
- Assignment Overview
- For next week

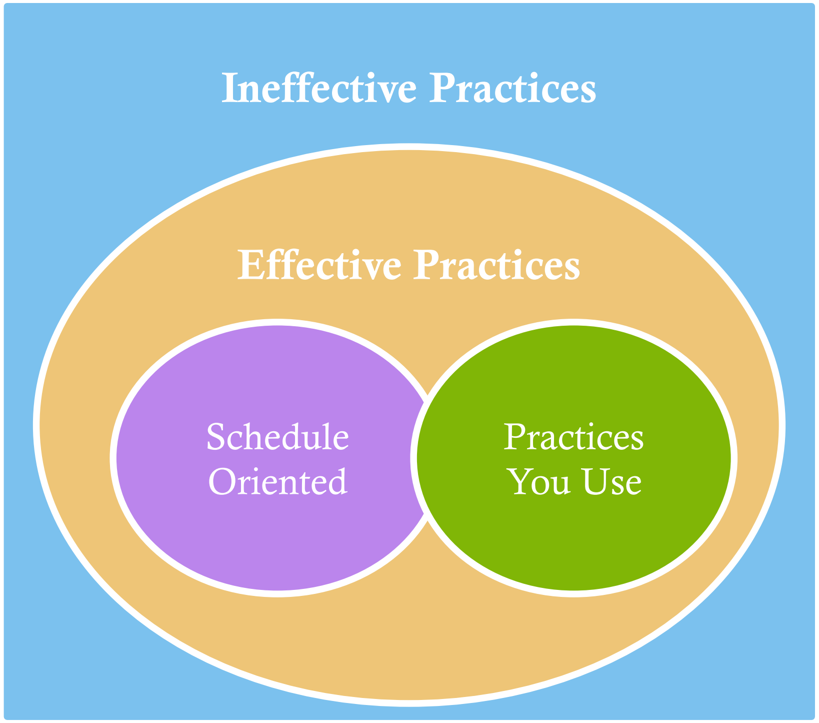
Review

McConnell, Chapters 1 & 2

Rapid Development

- "Any project that needs to emphasize development speed"
  - McConnell, pg.2

Software Practices



- McConnell, pg.3

The Four Pillars
of Rapid Development

Best Possible Schedule

1

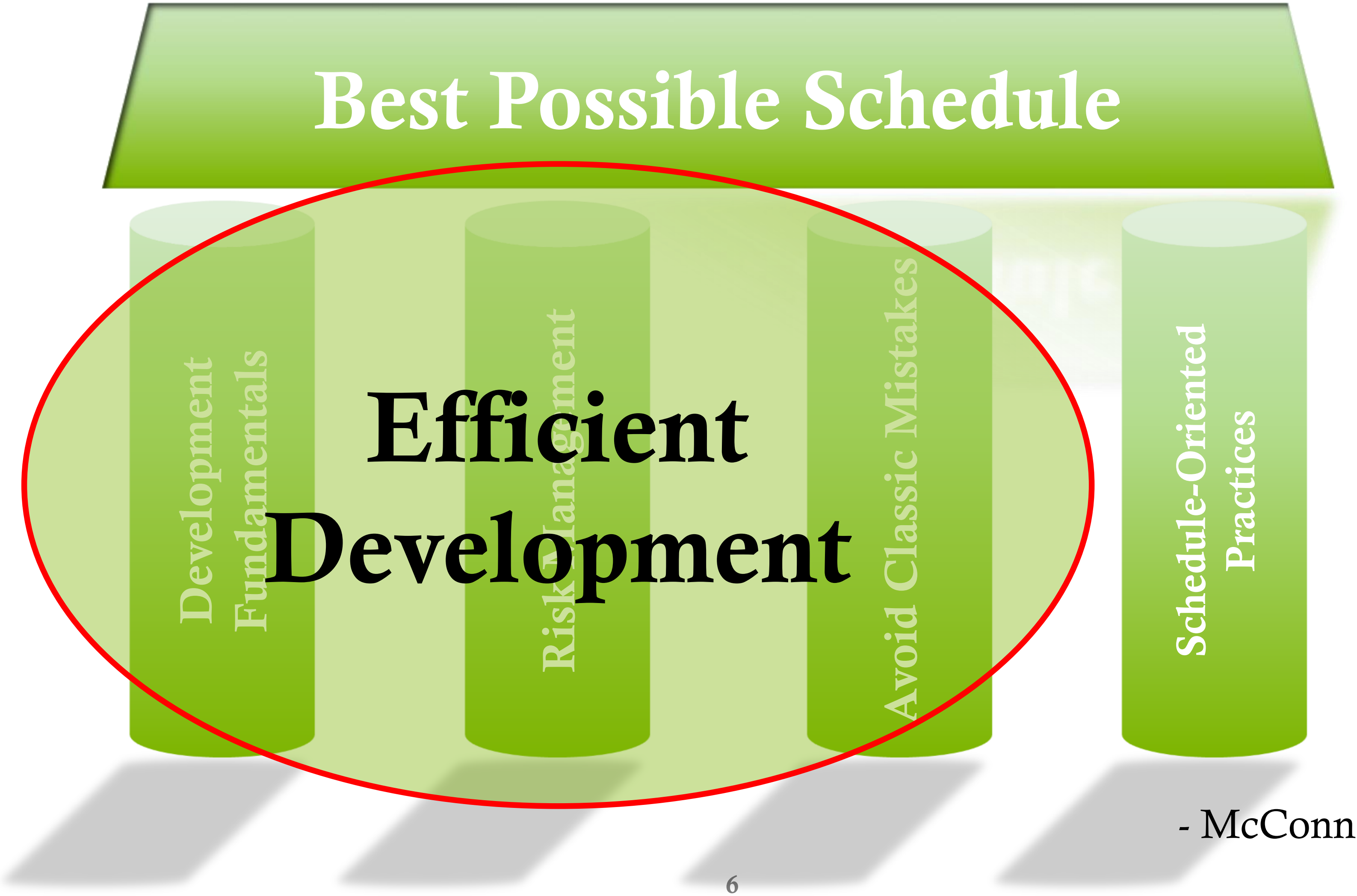
2

3

4

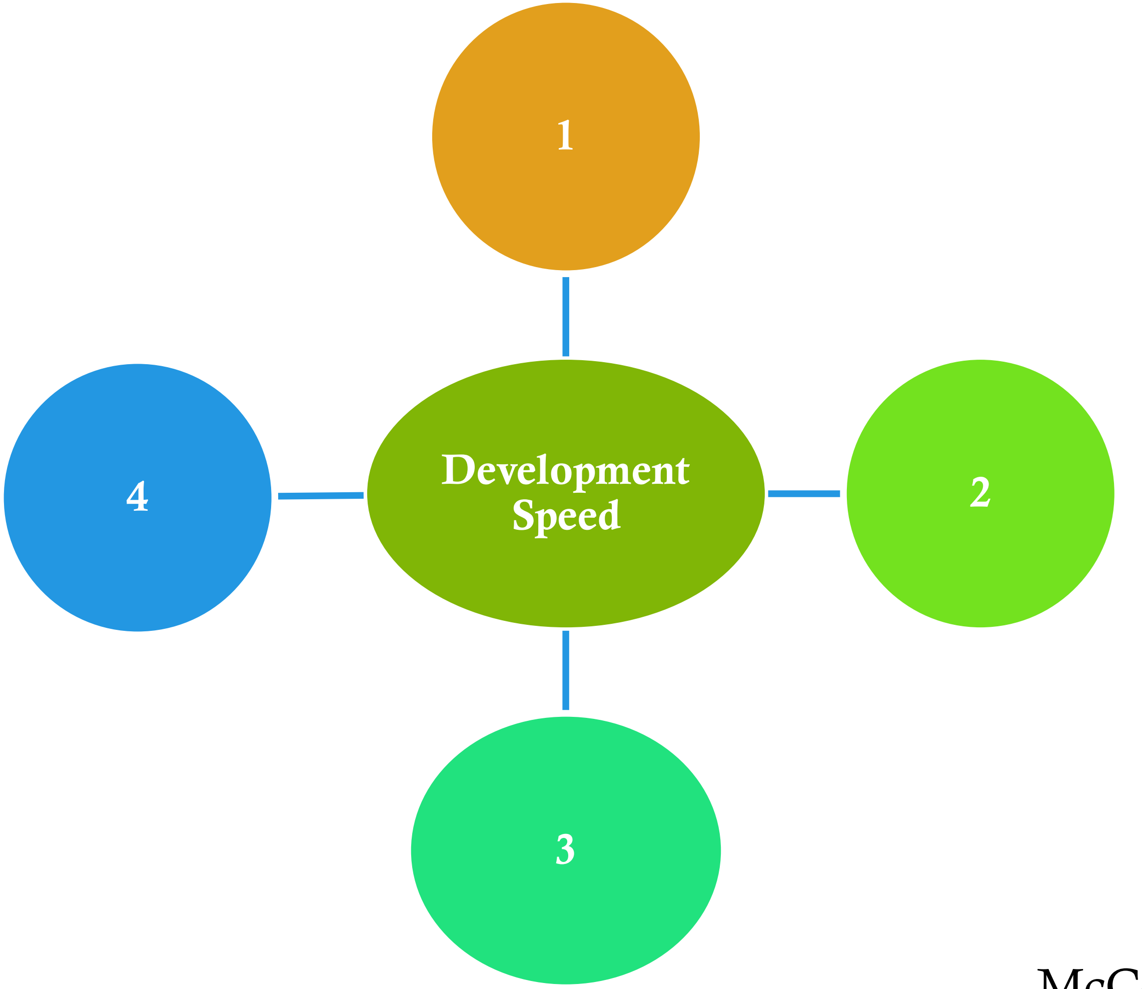
- McConnell pg.9

The Four Pillars
of Rapid Development



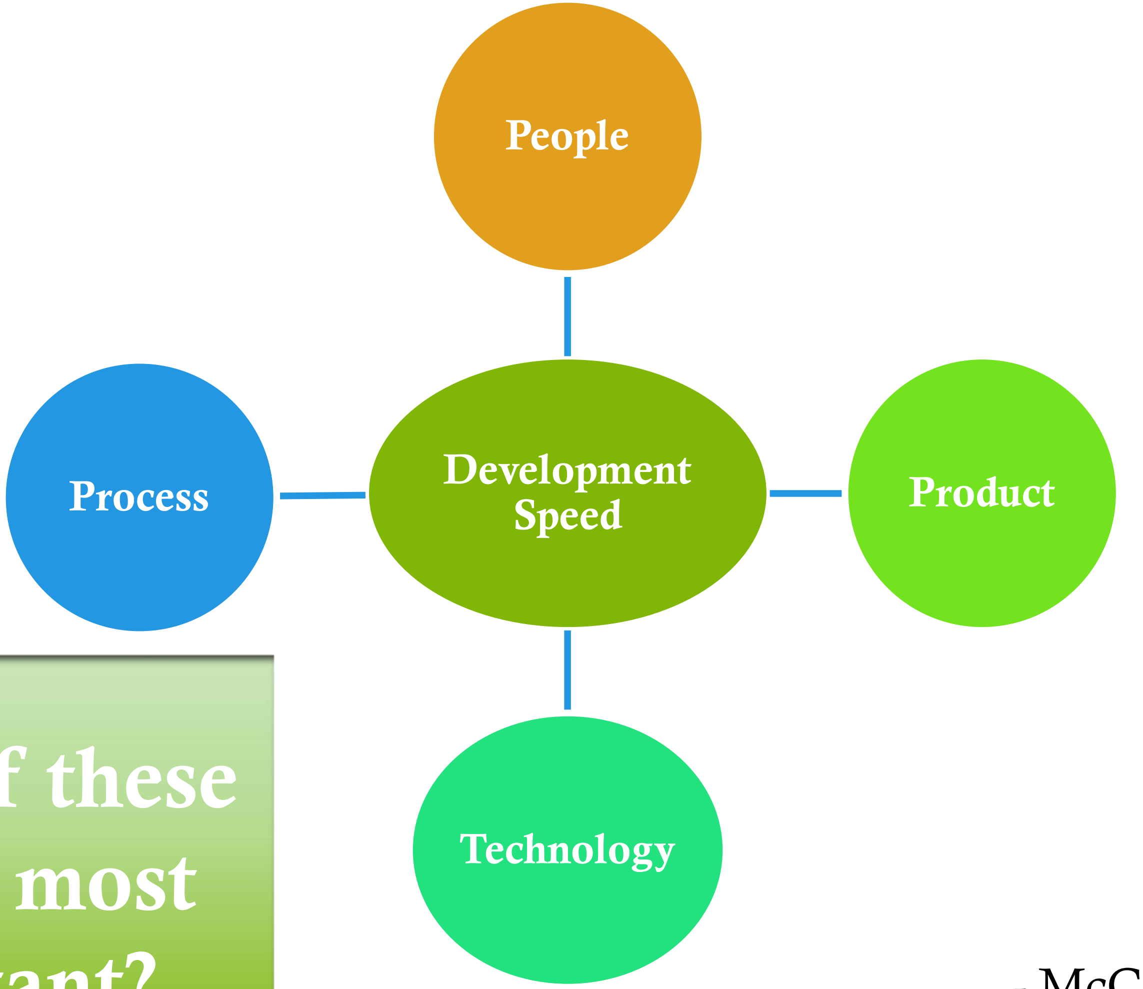
- McConnell pg.9

The Four Dimensions
of Development Speed



- McConnell pg.11

The Four Dimensions
of Development Speed



Which of these was the most important?

- McConnell pg.11

Industry Terminology

Industry Terminology

Quality Assurance (QA) – In this class we will refer to software testing and review as QA. In a system engineering context, it can refer to the group that ensures the agreed upon processes are being followed (i.e., requirements management, work management, etc.), often for certification with organizations such as ISO.

Verification and Validation – Two aspects of software testing.

- **Verification** – Does the software meet the specification (i.e., requirements, user stories).

- **Validation** – Does the software meet the intended purpose.

Industry Terminology

Technical Debt – Potential cost of re-work caused by choosing an easy or sub-optimal solution instead of a better or more robust solution that would take longer upfront to build.

Baseline – Reviewed and approved project or technical artifact, such as a plan, set of requirements or design. Change to a baseline is controlled through change management processes, such as a Change Control Board (CCB).

There could be Technical Baselines, Schedule Baselines, Financial Baselines, etc.

Classic Mistakes

McConnell, Chapter 3

Classic Mistakes

"The use of any specific best practice is necessary but not sufficient for achieving maximum development speed. Even if you do a few things right… you might still make a mistake that nullifies your productivity gains." McConnell, p.38

- What does he mean?
- What is a "best practice"?
- What makes a mistake "classic"?

Four Dimensions
elaborated

People

- "Peopleware"
- Staff selection
- Team organization
- Motivation

Process

- Rework avoidance
- QA
- Development fundamentals
- Risk management
- Resource targeting
- Lifecycle planning
- Customer Orientation

Product

- Product size
- Product characteristics

Technology

- Tools
- Languages

This is the list of things to make sure you do *WELL.*

- McConnell pg.11

The 36 Original Classic Mistakes

People-Related	Process-Related	Product-Related	Technology-Related
1. Undermined motivation 2. Weak personnel 3. Uncontrolled problem employees 4. Heroics 5. Adding people to a late project 6. Noisy, crowded offices 7. Friction between developers and customers 8. Unrealistic expectations 9. Lack of effective project sponsorship 10. Lack of stakeholder buy-in 11. Lack of user input 12. Politics placed over substance 13. Wishful thinking	14. Overly optimistic schedules 15. Insufficient risk management 16. Contractor failure 17. Insufficient planning 18. Abandonment of planning under pressure 19. Wasted time during the fuzzy front end 20. Shortchanged upstream activities 21. Inadequate design 22. Shortchanged quality assurance 23. Insufficient management controls 24. Premature or too frequent convergence 25. Omitting necessary tasks from estimates 26. Planning to catch up later 27. Code-like-hell programming	28. Requirements gold-plating 29. Feature creep 30. Developer gold-plating 31. Push me, pull me negotiation 32. Research-oriented development	33. Silver-bullet syndrome 34. Overestimated savings from new tools or methods 35. Switching tools in the middle of a project 36. Lack of automated source

This is the list of things we mess up again and again.

- McConnell pg.49

Exercise: Ch3 Case Study

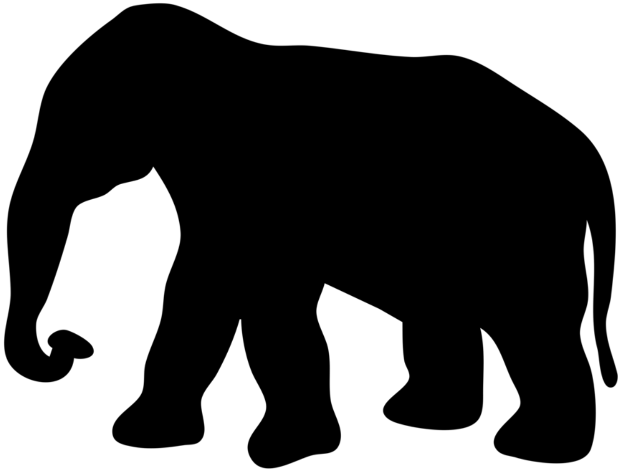
- Discuss in groups of 2 – 3, then we will debrief together
  - Which of these 36 Classic Mistakes (pp. 40-48) can you identify in the Chapter 3 Case Study?
  - Which one(s) had the biggest impact? Why?
  - How or when do you recognize these mistakes: the beginning, middle or end?
  - "Undermined motivation" is the first people-related mistake listed - what motivates you?

More Classic Mistakes (from Construx.com)

- Confusing estimates with targets
- Excessive multi-tasking
- Assuming global development has a negligible impact on total effort
- Unclear project vision
- Trusting the map more than the terrain
- Outsourcing to reduce cost
- Letting a team go dark (replaces the previous "lack of management controls")

Can you identify any additional Classic Mistakes from your experience?

Classic Mistakes



- Do you recognize any of the 36 classic mistakes from your experience?
- Why do we keep making these mistakes?
- Are any mitigated or even redundant now?
- How do **we** (you) avoid them in the future?

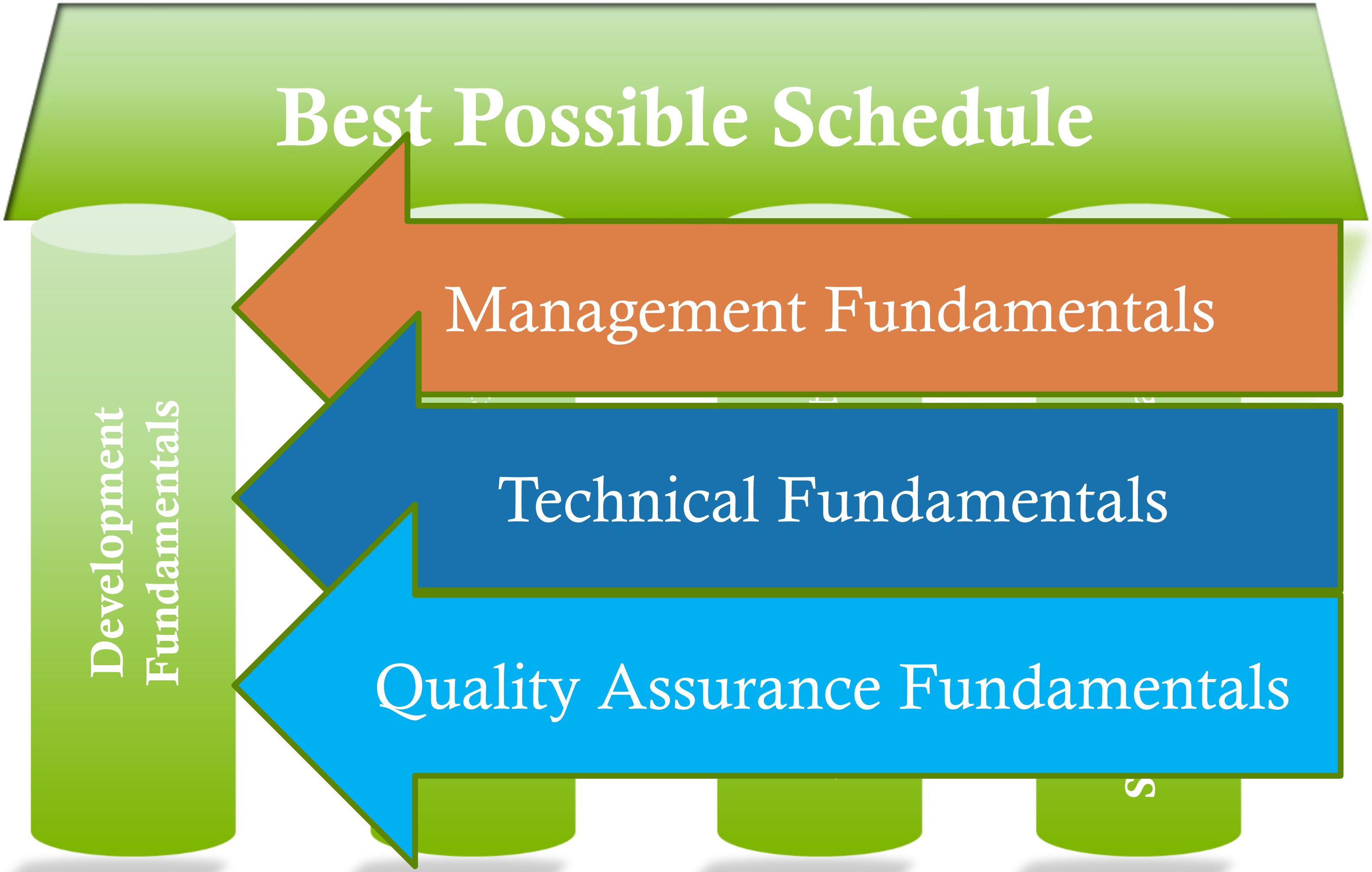
Mistakes in Industry

- 2015 HSBC, system failure
  - 275,000 transactions (wage payments) did not go through, right before a long weekend
- 2016 HSBC, 2 day service outage
  - "The frequency of these failures across the financial services sector suggests a systemic weakness in IT infrastructure."
- 2016 GM car recall, affecting 4 million vehicles
  - air bag software defect, found in 2014
  - sensing module enters test mode, air bags do not deploy

Development Fundamentals

McConnell, Chapter 4

Development Fundamentals



Best Possible Schedule

Development Fundamentals

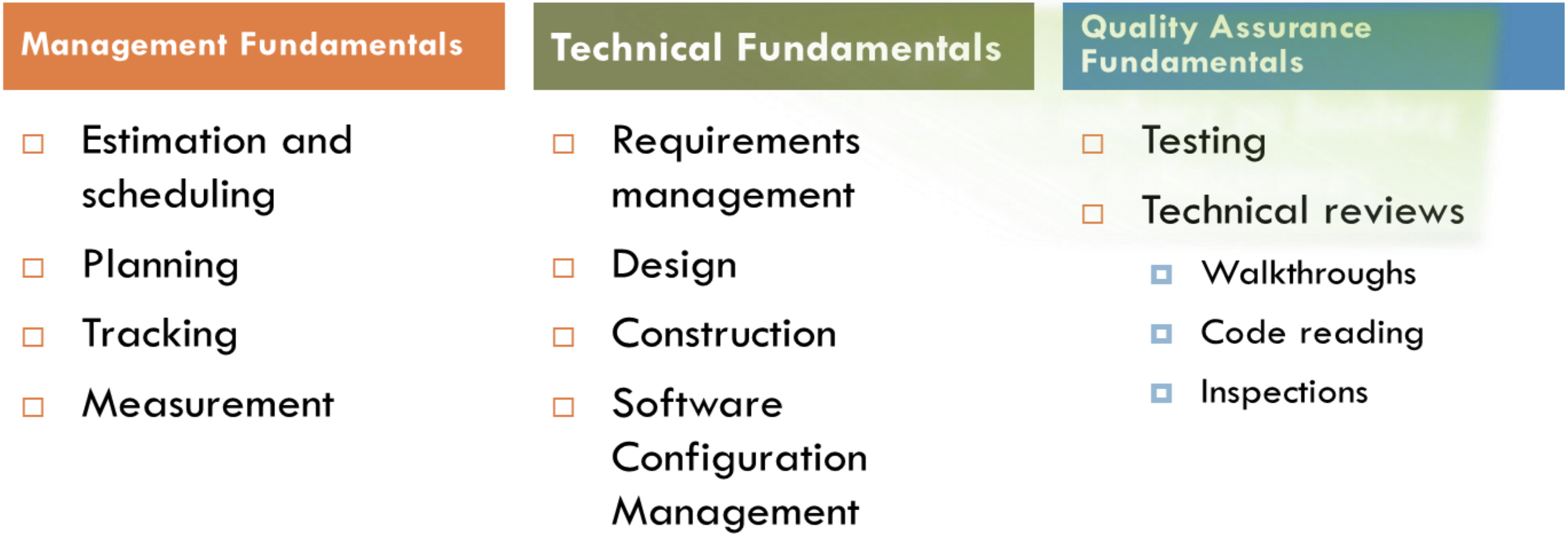
Management Fundamentals

Technical Fundamentals

Quality Assurance Fundamentals

Development Fundamentals

Which of these are handled well in your experience? Which one or two makes biggest positive impact to project success, and why?



Management Fundamentals

- Estimation and scheduling
- Planning
- Tracking
- Measurement

Technical Fundamentals

- Requirements management
- Design
- Construction
- Software Configuration Management

Quality Assurance Fundamentals

- Testing
- Technical reviews
  - Walkthroughs
  - Code reading
  - Inspections

How Important is QA?

- Ariane 5 “Cluster” in 1996
  - 10 years
  - $370M invested
  - Lost in 37 seconds
- *“The failure has become known as one of the most infamous and expensive software bugs in history.”*

Source: Wikipedia

What would be the biggest contributing factor to the QA failure?

Case Study:
compass Card Project

- What did we find out about the Compass project?
- What were the key points of the program, e.g. cost, duration, goal(s), etc.
- What has happened?

Compass Card Articles:

- http://www.vancouversun.com/news/TransLink+interim+defends+Compass+Card+delays/10942115/story.html#ixzz3mVvNtBN7
- http://www.theglobeandmail.com/news/british-columbia/metro-vancouvers-compass-transit-smart-card-system-to-run-in-fall/article25871062/
- http://thetyee.ca/News/2015/02/20/Compass-Card-Hires-Lobbyist/

Exercise: Compass Card Project

TOPICS

Management
pg.55-58

Technical
pg.60-66

Quality Assurance
pg.69-75

In groups of 2-3:

- Review the Dev Fundamentals topics and consider the Case Study
- What would you say about this case from the perspective of Dev Fundamentals and Classic Mistakes?
  - (1) Errors
  - (2) Things that went right
- How would you fix it?

Case Study:
Compass Card Project

- should never have promised… would be operating in three years (summer 2013)

- blamed a glitch with the "tap out" function at the end of a bus ride… field testing has shown the card readers are taking longer than anticipated to scan … up to several seconds from the target of 0.3 seconds

- push up the price of the system, which has already risen from $171.3 million to $194 million
    - http://www.vancouversun.com/news/TransLink+interim+defends+Compass+Card+delays/10942115/story.html#ixzz3mVvNtBN7

Case Study: Compass Card Project

- “fully deployed by this November…”
- “will launch after a phased rollout that has seen the system used by various users over its development”
- “The idea that you would stick with coins and pieces of paper is very 20th century”… TransLink is expecting that traditional options for paying for transit will exist for quite some time…
  - http://www.theglobeandmail.com/news/british-columbia/metro-vancouvers-compass-transit-smart-card-system-to-run-in-fall/article25871062/

Case Study: Compass Card Project

- San Diego-based Cubic was awarded the Compass project contract in late 2010, more than a year after it was announced jointly by the federal and B.C. governments as a $100-million project to stop fare evasion and enhance service planning

- Chicago Transit Authority experienced a bumpy launch for its Cubic-supplied Ventra fare payment system. In late 2013, it wanted a $1.2-million refund because of malfunctioning equipment
  - http://thetyee.ca/News/2015/02/20/Compass-Card-Hires-Lobbyist/

Your Practices

Individually, make a list of:

- What tools and/or techniques do you use for each of the topics below?
- Which have you found useful? Which are frustrating?

Management Fundamentals

- Estimation and scheduling
- Planning
- Tracking
- Measurement

Technical Fundamentals

- Requirements management
- Design
- Construction
- Software Configuration Management

Quality Assurance Fundamentals

- Testing
- Technical reviews
  - Walkthroughs
  - Code reading
  - Inspections

Management Fundamentals

Estimating and Scheduling

- Spreadsheets
- Historical data from accounting system

Planning

- MS Project
- Work Breakdown Structure (WBS)
- Custom spreadsheets

Tracking

- Change Requests/ Orders
- Status reports
- Work Package Status Report (WPSR)
- Completed stories

Measurement

- JIRA dashboards ie Kanban board
- WBS
- Charge codes or Timesheet system

Technical Fundamentals

Requirements Management

- Spreadsheet or document with numbered requirements
- Rational Requisite Pro
- User Stories

Design

- UML
- Standard templates
- Visio

Construction

- Various programming languages
- Various frameworks

Configuration Management

- Subversion and Git for code
- CMS for internal documents
- CM or DM department for formal deliverables

QA Fundamentals

Testing (manual)

- Spreadsheets or documents
- Requirements Tracebility Matrix (RTM)

Testing (automated)

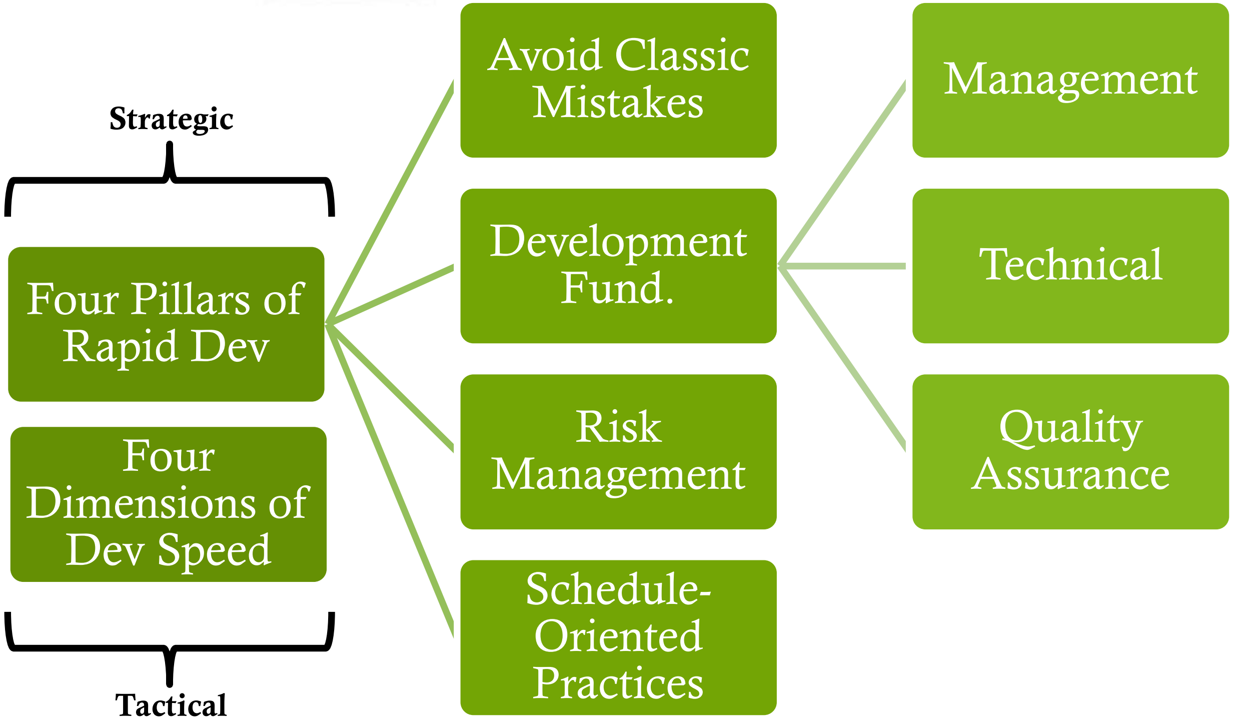
- Unit Tests
- Selenium
- Jenkins

Technical Reviews

- GitLab inline code reviews
- Checklists
- Formal sign-off on work packages
- JIRA workflows to ensure code reviews

Summary

Rapid Development Layers



Strategic

Four Pillars of Rapid Dev

Four Dimensions of Dev Speed

Tactical

Avoid Classic Mistakes

Development Fund.

Risk Management

Schedule-Oriented Practices

Management

Technical

Quality Assurance

The Bottom Line

So what?

- Who is responsible for sorting all of this out?
- What is a necessary and what are the sufficient conditions?
- Is it realistic – either the picture McConnell paints, or the prescriptions that follow?

Assignments Overview

Assignment 1

- Individual
- Due: end of week 5
  Feb 11th, midnight
- Research and compare two development approaches to Rapid Development
- Details posted now on D2L

Assignment 2

- Groups of 2
- Due: TBD
  (after midterm)
- Form pairs, later
- Research and compare two Best Practices
- 15 minute in-class presentations
- Details will be posted later on D2L

Assignment 3

- Different groups of 2
- Due: in-class, week 13

In Class:

- Form pairs
- Case Study provided
- Analysis to be worked on and submitted by end of class

For Next Week (week 3)

To Do before next class:

- Read McConnell, chapter 5 on Risk Management
- Look up some information on the rollout of Healthcare.gov
  - We are interested only in the website rollout, not the policy

Comp8081

end of Week 2

Donna Turner