Outline

• Why CMM matters
• Historical background
• Sports analogy
• What is CMM?
• How you can use CMM
• Details about CMM
• Problems with CMM
Threads from this course

- Iteration in design process
- Choice of process for development
- Is software different from other (physical) engineering fields?
Why CMM matters…

• It is the most widespread and detailed software development model
• It is a standard for much DoD work, which is a lot of software projects
• It is being used by many non-DoD businesses
• It is widely criticized, and has inspired several anti-CMM models
• Your tax dollars paid for it
Background

• Begun in 1986 by DoD to help improve government software contractors.
• Work started at Mitre, then at Software Engineering Institute (SEI) at Carnegie Mellon Univ.
• Watts Humphrey was initial author, then Mark Paulk, Bill Curtis, and others.
• Borrows heavily from general Total Quality Management (TQM) and work of Philip Crosby.
Background

• Under active development for 15+ years, ongoing…
• Has gained significant interest among non-DoD software vendors.
• All documents are public, and many are free.
• Known as SW-CMM or CMM.
SW development and baseball

• What happens when a ball is hit to a Little League team?
  • Everyone runs around at random.
  • They might do the right thing, or they might not.
  • The next time the ball is hit in the same place, they may do something different.

• What happens when a ball is hit to a professional team?
  • Everyone moves in a coordinated fashion, based on practicing that play many times.
  • Sometimes they fail to make the right play, but they almost always try to do the right thing.
SW development and baseball

• What happens when the team loses a star player?
  - Little League team gets much worse.
  - Professional team often has someone waiting to fill in.

• Self-improvement after a bad play…
  - Little League players don’t know what went wrong, or they blame each other.
  - Professional teams discuss their play and look for ways to improve. "The next time there is an infield hit with 2 outs, let’s do this instead."
SW development and baseball

• A professional baseball team is more "mature" than a Little League team (not referring to age).

• A professional team has self-perpetuating quality. They
  • Make good plays
  • Develop new players like themselves
  • Find ways to make better plays
What is CMM?

• In the same way, high-quality SW organizations are different from low-quality orgs.
• CMM tries to capture and describe these differences.
• CMM strives to create software development organizations that are “mature”, or more mature than before applying CMM.
• Describes five levels of SW process maturity.
• Includes *lots* of detail about each level – we will look at some of it.
How to use CMM

• Hire an officially certified CMM Assessor to conduct a formal evaluation.
  • To win government software contracts.
  • To find high-quality software subcontractors. (SA-CMM)
  • For pure development shops, to impress clients with your quality. (India)

• Send your own people to official CMM training, then conduct internal assessments.
  • For a large organization where software process improvements have a big payoff.
How to use CMM

• Use CMM as a set of suggestions and apply as you see fit.
  • Every other software development organization, of all sizes.
Summary of levels

• Level 1 – Initial. Anything at all. Ad-hoc and chaotic. Will have some successes, but will also have failures and badly missed deadlines.

• Level 2 – Repeatable. SW processes are defined, documented, practiced, and people are trained in them. Groups across an organization may use different processes.
Summary of levels

• Level 3 – Defined. SW processes are consistent and known across the whole organization.
• Level 4 – Managed. SW processes and results are measured quantitatively, and processes are evaluated with this data.
• Level 5 – Optimizing. Continuous process improvement. Experimenting with new methods and technologies. Change processes when find something that works better.
Level 1 – Initial

• Team tackles projects in different ways each time
• Can have strong successes, but may not repeat
• Some time/cost estimates are accurate, many far off
• Success comes from smart people doing the right things
• Hard to recover from good people leaving
• Frequent crises and "firefighting." (Many believe this is standard for SW development. CMM says NO.)
• Most SW development organizations are Level 1.
• Estimating curve, process diagram.
Level 2 – Repeatable

• Key areas
  • Requirements management
  • Software project planning
  • Project tracking and oversight
  • Subcontracts management
  • Quality assurance
  • Configuration management

• Usually takes 18+ months. (Some ask for Level 1.5.)

• Estimating curve

• Process diagram
Level 3 – Defined

• Key areas. Level 2, plus…
  • Organization-wide process focus
  • Organization-wide process definition
  • Training program in above
  • Integrated software management (above applied per project)
  • Software product engineering (coding, etc.)
  • Inter-group coordination
  • Peer reviews

• Estimating curve
• Process diagram
Level 4 – Managed

• Key areas. Level 3, plus…
  • Quantitative process management (data gathering)
  • Quality management (data-driven quality improvement)
• Estimating curve
• Process diagram
Level 5 – Optimizing

• Key areas. Level 4, plus…
  • Defect prevention
  • Technology change management (bring in new methods)
  • Process change management (improve processes)
• Estimating curve
• Process diagram
Structure of the Levels (OPT)

- Maturity Levels (5)
- Key Process Areas (2-7)
  - Goals (2-4)
  - Commitment to perform
  - Ability to perform
  - Measurement and analysis
  - Verification
  - Activities performed
    - Key practice #1
    - Key practice #2
    - ...

- Main idea: Saying you will do something is not enough.
Observation:

• CMM contains a lot of verbiage.
• Can seem like just a word game, which it might be to some extent.
• If you get beyond the verbiage however, CMM does describe some important methods for running software projects.
“Maturity suite” (OPT)

- Personal Software Process (PSP)
- Team Software Process (TSP)
- People CMM (P-CMM)
- Software Acquisition CMM (SA-CMM)
- System Engineering CMM (SE-CMM)
- Integrated Product Development CMM (IPD-CMM)
- CMM Integration (CMMI): SW + SE + IPD
Problems with CMM

- It is a goal, not a method
- Being used just as stamp of approval
- Doesn’t say anything about software!
- Doesn’t help in a crisis
- Only for repetitive tasks
CMM is a goal, not a method

- Organizations often look to CMM as a method or formula for improvement
  Tempting to want easy answers.
- CMM is actually a management framework, with many details left out
  Example: “You must have peer reviews.” But how should the reviews be run?
- If you say, "we do CMM just like the book", you aren't doing CMM
- To use CMM, you have to think
  You must be flexible, be creative, and integrate the goals of CMM with the realities of your business.
CMM is a goal, not a method

• Related to this point, some organizations are rigid about mandating CMM to their employees

• This has given CMM a bad reputation as an onerous, inflexible method

• The problem here (usually) is misunderstanding CMM, not its rigidity
Becoming stamp of approval

• Some organizations want CMM only as a stamp of approval, without a high-level commitment to process improvement or quality
• Want to find easiest way to get certified as Level 2 (or 3) without really changing
• I talked to the first CMM assessor in the world. She was tired and disillusioned. Why?
  • She wanted companies to say, “Let’s work together to improve our software processes.”
  • Instead, they say, “Just tell us what to do to get Level 2, so we can get back to work.”
Not about software!

• CMM is, by design, a model for *managing* software projects

  They claim most software failures are due to management problems rather than technical barriers (I agree)

• But CMM goes too far in this direction

• An organization could write lousy code (consistently) and be rated highly by CMM

• This is counter-intuitive, since good code is the goal of software development
Does not help in a crisis

• CMM does not help projects that are in crisis right now
  Trying to apply it then would make things worse
• Unfortunately, this is often when companies look for help with their software processes
• Analogy: For long-term cardiac health…
  Eat lots of fruits and veggies, quit smoking, get exercise, reduce stress
• But suppose you are having a heart attack now. Will these actions help?
  It is too late for preventive measures. You need some other kind of intervention.
Only for repetitive tasks

• CMM is based on re-using past results for future software projects
  In management activities, quality measurements, development processes
• But, this only makes sense for relatively repetitive projects
  Example: MS Word team creating V7 after V1-6
• To create brand-new software of unknown size, with unknown hurdles, using human creativity, CMM is clearly not the right model.
Official CMM information

- [Software CMM home, summary paper, detailed paper](#)
- [Using CMM in small organizations](#)
- [Software Acquisition CMM home, paper](#)
- [Team Software Process](#)
- [Personal Software Process](#)
- [System Engineering & Product Development CMMs](#)
- [CMM Integration](#)
- [Comparison of CMM to ISO 9001](#)
Other authors about CMM

- **The Essence of the CMM**, Judy Bamberger. Computer, 6/97. Worth reading, by one of the authors of CMM.
- **The Immaturity of CMM**, James Bach. American Programmer, 9/94. This article, and the next, articulate common complaints about CMM.
- Agile software development (anti-CMM): [AgileManifesto.org](http://AgileManifesto.org) and [AgileAlliance.org](http://AgileAlliance.org).