



Microsoft

Visual Studio

Team System

Stretching Agile to Fit CMMI Level 3

– the story of creating MSF for CMMI Process Improvement at Microsoft

David J. Anderson, Microsoft Corporation

Our initial View

- **Defense Contractor business**
 - ▶ **Life critical**
 - ▶ **Regulated**
 - ▶ **Bureaucratic**
- **CMM devised for assessment of Air Force contractors**
- **Low trust environment**
 - ▶ **High audit requirements**
 - ▶ **Lots of documentation – waste?**

Deming 14 Points

- Cycle – PDCA (Plan, Do, Check, Act)
- 3. Cease dependence on quality control to achieve quality, instead focus on quality assurance throughout the lifecycle.
- 4. Build trust and loyalty with suppliers
- 6. Training on the job
- 7. Leadership
- 8. Drive out fear
- 9. Break down barriers between departments
- 12. Remove barriers to pride of workmanship, focus management on quality rather than production numbers

Deming Epiphany

- CMM Levels based on Crosby's Manufacturing Maturity Model
- But underlying principles of CMM aimed at Deming's quality assurance method
- Level 2 thru 4 for elimination of special cause variation
- Level 5 for continuous improvement through reduction of common cause variation
- Deming thinking is agile, so why can't a CMMI method be agile?

Seeing CMMI as non-agile

- Project Planning (PP) 2.1 asks us to *“identify task dependencies”* and expects a *“Critical Path Method (CPM) or Program Evaluation and Review Technique (PERT)”* [Chrissis 2003] chart as a result.
- PP 1.1 sub-practice 2 asks us to *“Identify the work packages in sufficient detail as to specify estimates of project, tasks, responsibilities and schedule”* and goes on with *“The amount of detail in the WBS at this more detailed level helps in developing realistic schedules, thereby minimizing the need for management reserve.”*

More non-agility

- **Project Monitoring and Control (PMC) 1.1** asks us to compare *“actual completion of activities and milestones against the schedule documented in the project plan, identifying significant deviations from the schedule estimates in the project plan.”*
- **Technical Solution (TS) 2.2** asks us to *“establish a technical data package”* and suggests artifacts like *“product architecture description, allocated requirements, product component descriptions, product characteristics, interface requirements, conditions of use”*
- **400 doc types, 1000 artifacts for an appraisal [Ahern 2004]**

CMMI Model

- Inspired by Deming's Theory of Profound Knowledge and continuous improvement through quality assurance
- 5 level model based on Philip Crosby's Manufacturing Maturity Model
 - ▶ Issue – too much Crosby conformance to plan and specification slipped into CMMI literature

What we achieved...

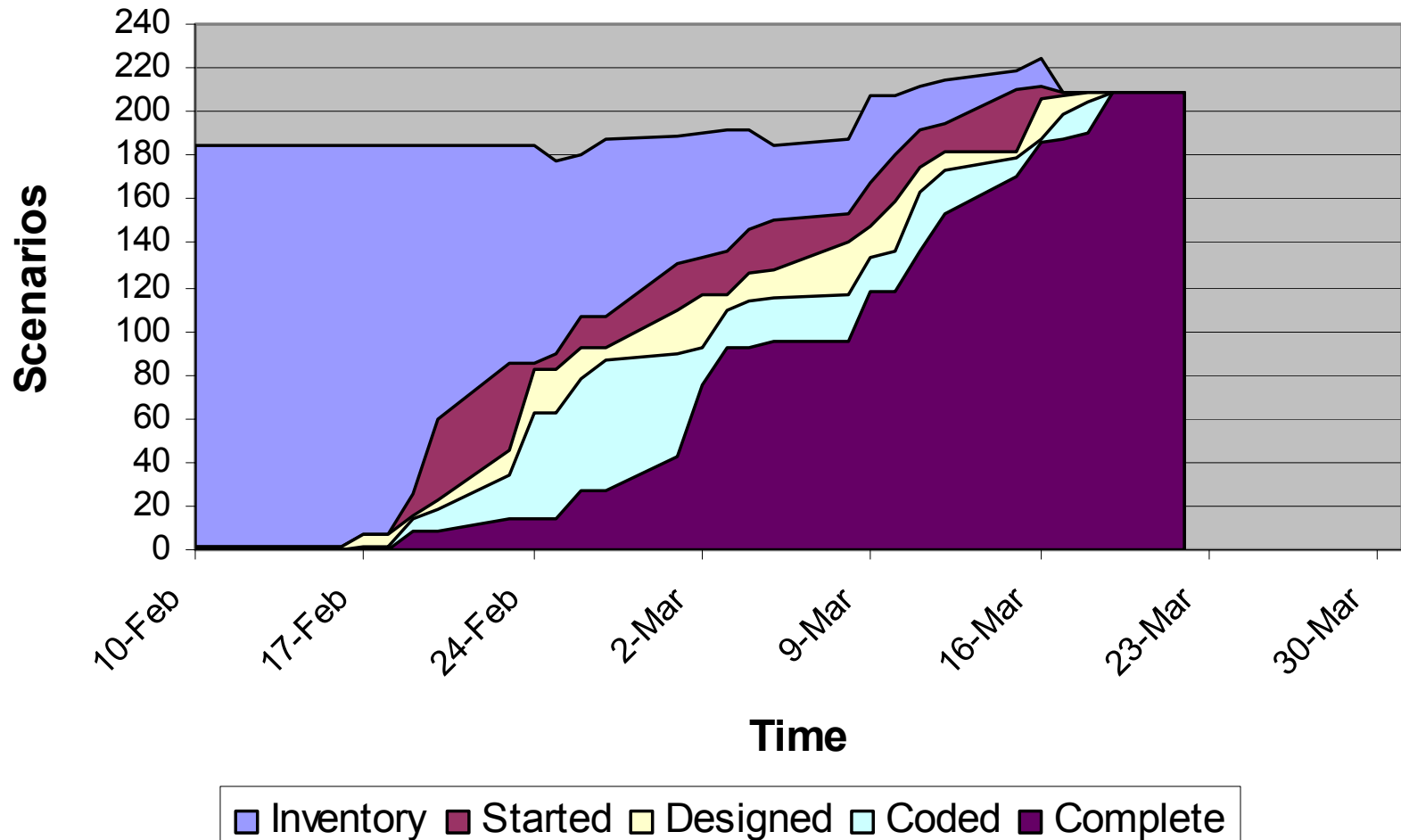
- Coverage of 20 out of 25 process areas
 - ▶ Omitted SAM, ISM, OEI, OPF, OT
- Footprint only 150% bigger than MSF for Agile Software Development
- Approximately 200 activities
- Only 50 documents
- Supported by 50 automated queries and reports

Iterative Project Planning

- **Loose approximate project plan**
 - ▶ End-to-end scenarios broken out into iterations, i.e. a project plan is a loose plan for a set of iterations
- **Postponed iteration planning**
 - ▶ Functionality only locked at iteration start
 - ▶ Scenario-based
 - ▶ Prioritized by customer value
- **All estimates remain approximate – within some band of variation**

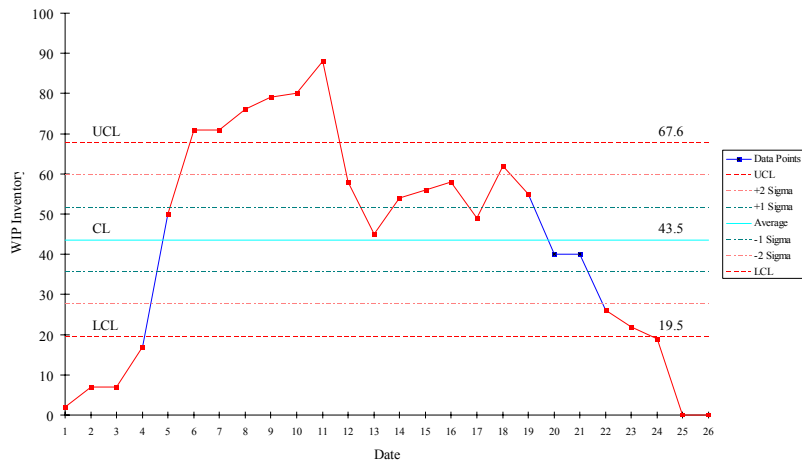
Cumulative Flow Chart

Iteration Cumulative Flow

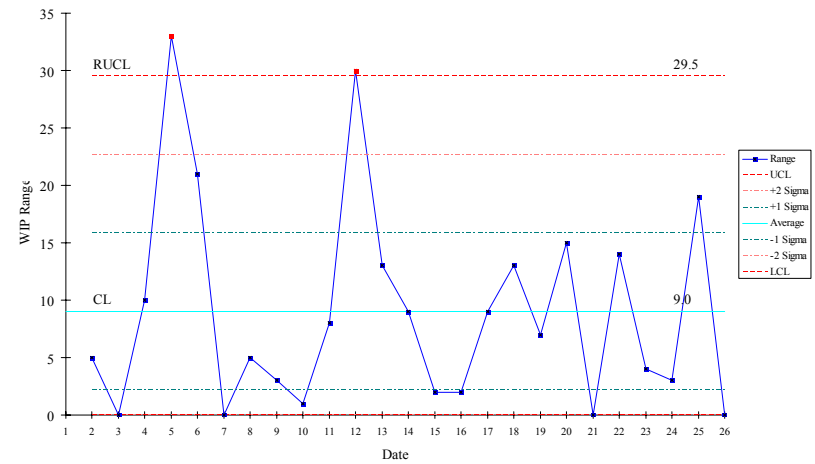


Work in Progress Control Charts

WIP Control Chart

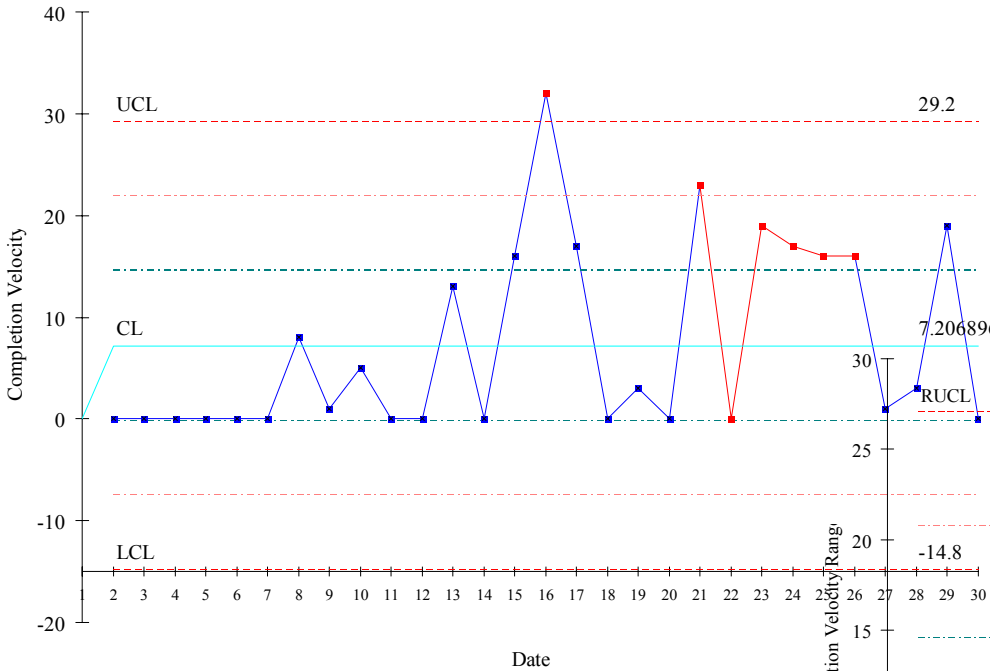


WIP Range Chart

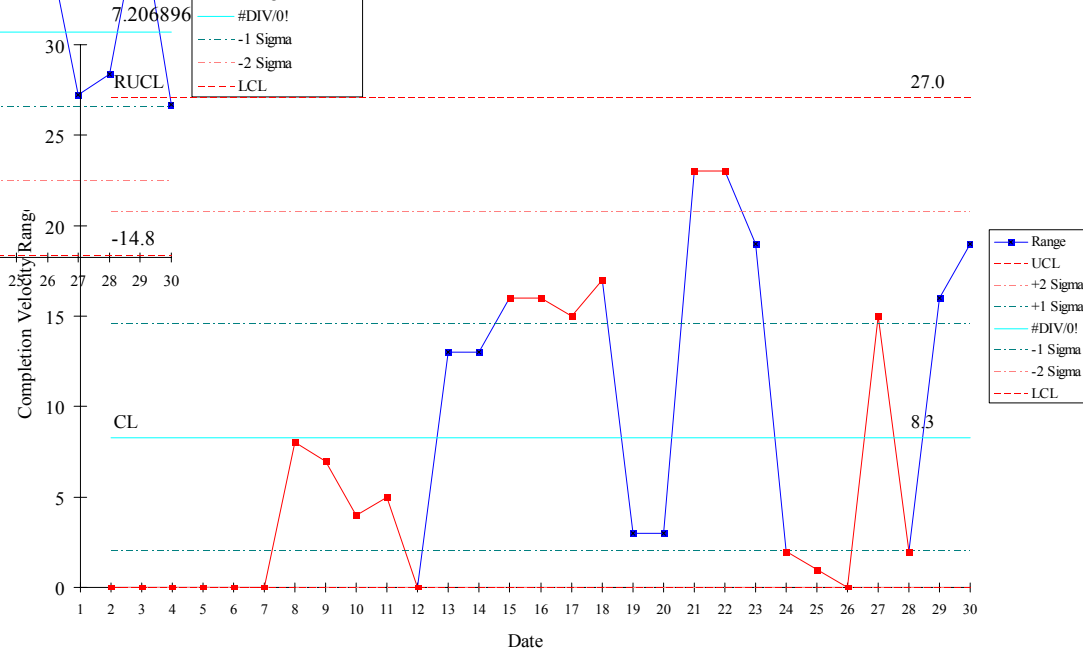


Velocity Control Charts

Completion Velocity Chart



Completion Velocity Range



Issues and Blocked Work

Issues and Blocked Work Items

Report generated: 11/04/2004 11:25 AM by jackerman

Description...

[View report documentation](#)

