High Maturity Practices and Lifetime Warranty Against Software Defects

A Disruptive Innovation?

ASQ Software SIG
April 23, 2013
Disruptive Innovation

Disruptive innovation, a term of art coined by Clayton Christensen, describes a process by which a product or service takes root initially in simple applications at the bottom of a market and then relentlessly moves up market, eventually displacing established competitors.

Software Engineering

Persistent Problems

1. Exponential rise in cybersecurity vulnerabilities due to defective software
2. Unacceptable cost, schedule, and quality performance of legacy systems modernization and Enterprise Resource Planning (ERP) projects
3. Cost of finding and fixing software bugs (i.e. scrap and rework) the number one cost driver in software projects
4. Arbitrary and unrealistic schedules leading to a culture of “deliver now, fix later”
5. Inability to scale software engineering methods even for medium size systems
6. Understanding the impact of variation in individual productivity
7. Absence of workplace democracy and joy in work
The Top 15 U.S. Software Cost Drivers in Rank Order Circa 2009

1. The cost of finding and fixing bugs
2. The cost of cancelled projects
3. The cost of producing paper documents and English words
4. The cost of recovery from security flaws and attacks
5. The cost of requirements changes during development
6. The cost of programming or coding
7. The cost of customer support
8. The cost of meetings and communication
9. The cost of project management
10. The cost of application renovation
11. The cost of innovation and new kinds of features
12. The cost of litigation for cancelled projects
13. The cost of training and learning software applications
14. The cost of avoiding security flaws
15. The cost of acquiring reusable components

Capers Jones, STN 13-1 April 2010: Software Quality, Reliability, and Error Prediction
Cybersecurity Vulnerabilities

"The next Pearl Harbor we confront could very well be a cyber attack that cripples our power systems, our grid, our security systems, our financial systems, our governmental systems,"

Leon Panetta, June 2011, before the Senate Armed Services Committee
# Classes of Problems

<table>
<thead>
<tr>
<th>Wicked Problems</th>
<th>Tame Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Messy</td>
<td>• Completely understood</td>
</tr>
<tr>
<td>• Ill-defined</td>
<td>• Crisply defined</td>
</tr>
<tr>
<td>• Open to multiple</td>
<td>• Fixed through complex technical solutions</td>
</tr>
<tr>
<td>interpretations</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

- Poverty
- Obesity

**Examples**

- Putting a man on the moon
- Devising a cure for diabetes

[http://www.newyorker.com/online/blogs/comment/2012/06/something-wicked-this-way-comes.html#ixzz25Lb6mG32](http://www.newyorker.com/online/blogs/comment/2012/06/something-wicked-this-way-comes.html#ixzz25Lb6mG32)
Solutions

Tame problems are solvable
Solutions to tame problems either work or they don’t
Solutions to wicked problems, by contrast, are only better or worse
Trade-offs are unavoidable
Unanticipated complications and benefits are both common
Difficult or impossible to learn by trial and error
No solution to a wicked problem is ever permanent or wholly satisfying, which leaves every solution open to easy polemical attack

http://www.newyorker.com/online/blogs/comment/2012/06/something-wicked-this-way-comes.html#ixzz25Lb6mG32
Growth Industries

Information Assurance
Certification & Accreditation
Testing
Code Analyzers
Certifications
  PMP, CMMI, Scrum
CMMI, A Procedural Process - 1

CMMI defines characteristics of an effective process

Sets organizational standards and baseline procedures for use by all projects in an organizational unit

Organizations use CMMI to enforce repeatable processes and work in a predictable way
CMMI, A Procedural Process - 2

CMMI is extremely helpful in stabilizing an organization and getting a level of statistical control

CMMI levels build the foundation needed for real improvement

CMMI level 5 is the beginning and not the end
CMMI Improves Quality

<table>
<thead>
<tr>
<th>CMM Level</th>
<th>Defect Potential per Function Point</th>
<th>Defect Removal Efficiency</th>
<th>Delivered Defects per Function Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMM 1</td>
<td>5.50</td>
<td>73.00%</td>
<td>1.49</td>
</tr>
<tr>
<td>SEI CMM 2</td>
<td>4.00</td>
<td>90.00%</td>
<td>0.40</td>
</tr>
<tr>
<td>SEI CMM 3</td>
<td>3.00</td>
<td>95.00%</td>
<td>0.15</td>
</tr>
<tr>
<td>SEI CMM 4</td>
<td>2.50</td>
<td>97.00%</td>
<td>0.08</td>
</tr>
<tr>
<td>SEI CMM 5</td>
<td>2.25</td>
<td>98.00%</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Capers Jones, STN 13-1 April 2010: Software Quality, Reliability, and Error Prediction
Issues with Procedural Processes

Procedural processes rely on artifacts to make sure process is being followed

Artifacts are produced by organizational bureaucracy and not the developers

Artifacts may have no relationship to the actual work being done

Easier to pass appraisals than to change engineering behavior
The Real Question

Whose Process Is It?

“If our methods do not serve the individual professionals, they will not endure”
Agile Manifesto

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan
Agile Manifesto
Assumptions - 1

Requirements are not knowable in advance

Less than 30% of a customer's initial requirements can be captured in the form of written documentation

Requirements change throughout the project

Planning and tracking project progress is non-value adding

More than 65% of requirements are never used at all

Agile Manifesto
Assumptions - 2

Big up-front design causes more problems than it solves

Creates larger attack surfaces vulnerable to security incidents

Large projects are unsuccessful or canceled

Requirements defects are #1 reason projects fail

DoD and Agile

4. Enable Agile IT

- **Active user involvement**
  to prioritize requirements and provide responsive feedback during development

- **A different approach to project management**
  - Small, dynamic, and empowered teams

- **Roadmaps and architectures**
  align agile increments into larger capabilities

- **Deliver usable capabilities**
  to users every 6-12 months

- **Small scoped releases**
  responsive to changes in ops, tech, budget...

- **Streamlined contracting processes**
  leveraging existing contract vehicles for rapid Task/Delivery Order execution

- **Leveraging common infrastructure platforms, standards, and interfaces**

- **Integrated test and evaluation, certifications during development**
  leveraging common test infrastructure, automated tools
Agile Sprints

Example
  Two year modernization project to deliver 600,000 lines of code
  Two week sprints
    Working software 12,000 LOC every two weeks

Issues
  Management overhead
  Customer acceptance testing time
  Defect density in delivered product
  Work-Life balance
  “Deliver now, Fix later”? 
Agile Manifesto 2.0?

We value **doing** more than **Plan, Do, Check, Act**
We value **guesswork** more than **facts**
We value **speed** more than **optimum results**
We value **accumulating technical debt** more than institutionalizing **disciplined approach**

If customers want it in the worst way, that’s how we will deliver it
Software Engineering
Persistent Problems

1. Exponential rise in cybersecurity vulnerabilities due to defective software
2. Unacceptable cost, schedule, and quality performance of legacy systems modernization and Enterprise Resource Planning (ERP) projects
3. Cost of finding and fixing software bugs (i.e. scrap and rework) the number one cost driver in software projects
4. Arbitrary and unrealistic schedules leading to a culture of “deliver now, fix later”
5. Inability to scale software engineering methods even for medium size systems
6. Understanding the impact of variation in individual productivity
7. Absence of work place democracy and joy in work
The Way Forward -1

CONNECT THE DOTS

AGILITY, QUALITY, INNOVATION,

AND

JOY IN WORK
The Way Forward -2

Unhappy people rarely do quality work
Without quality, agility is in name only
Quality without numbers is just talk
Culture of innovation is possible only in
   environments where people are respected,
   self-aware and practice agility with discipline
We can get there by applying the principles of
   Deming, Drucker, and Humphrey to software
   technical work
Dr. Deming

Constancy of purpose brings innovation
Various components should work together for optimization of profit and joy in work
Rewards motivate people to work for the rewards
Extrinsic motivation leads to the destruction of the individual
Remove barriers that rob people of pride of workmanship
Numerical goals set for other people, without a road map to reach the goal, have effects opposite to the effects sought
Dr. Drucker

Three-step process – recording time, managing time, consolidating time – is the foundation for effectiveness.

What motivates technical professionals is what motivates volunteers, who have to get more satisfaction from their work than paid employees, precisely because they do not get a paycheck.

That one can truly manage other people is by no means adequately proven.
Watts Humphrey

Agility

Quality is more important than schedule

“In today’s software marketplace, the principal focus is on cost, schedule, and function; quality is lost in the noise. This is unfortunate since poor quality performance is the root cause of most software cost and schedule problems.”

Professional Discipline

Not authoritarian, parental; more like athletes, concert artists, surgeons

Includes effective defect management, comprehensive planning, and precise project tracking and reporting

“If our methods do not serve the individual professionals, they will not endure”
Koestenbaum

Don’t think of business as a life without greatness

Unless the distant goals of meaning, greatness, and destiny are addressed, we can’t make an intelligent decision about what to do tomorrow morning – much less set the long-term strategy of the company

First decision must be to commit to an ethical world, a civilized existence, a moral order

Nothing is more practical than for people to deepen themselves

www.pib.net
Constancy of Purpose
Schedule Performance

AIS Schedule Deviation Control Chart - Development Phases

% Deviation

Date of Project Phase Start

One Standard Deviation
Average

CMM
PSP/TSP

112.3 %
36.8 %
10.5 %

1/88 1/89 1/90 1/91 1/92 1/93 1/94 1/95 1/96 1/97 1/98 1/99 1/00 1/01 1/02 1/03 1/04 1/05 1/06 1/07 1/08 1/09 1/10 1/11 1/12
Constancy of Purpose
Cost Performance
## Performance Metrics That Matter Benchmarking

<table>
<thead>
<tr>
<th>Metric</th>
<th>Industry Average</th>
<th>AIS Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule deviation</td>
<td>&gt;50%</td>
<td>&lt;10%</td>
</tr>
<tr>
<td>No. of defects in delivered product 100,000 LOC</td>
<td>&gt;100</td>
<td>&lt;15</td>
</tr>
<tr>
<td>% of design and code inspected</td>
<td>&lt;100</td>
<td>100</td>
</tr>
<tr>
<td>Time to accept 100,000 LOC product</td>
<td>10 Months</td>
<td>5 Weeks</td>
</tr>
<tr>
<td>% of defects removed prior to system test</td>
<td>&lt;60%</td>
<td>&gt;85%</td>
</tr>
<tr>
<td>% of development time fixing system test defects</td>
<td>&gt;33%</td>
<td>&lt;10%</td>
</tr>
<tr>
<td>Cost of quality</td>
<td>&gt;50%</td>
<td>&lt;35%</td>
</tr>
<tr>
<td>Warranty on products</td>
<td>?</td>
<td>Lifetime</td>
</tr>
</tbody>
</table>
Competitive Game Changers?

Lifetime warranty against software defects
Firm Fixed Price with performance guarantees
Precise and accurate tracking with ability to detect one-day slip in schedule
Level 5 behavior - Organization, Team, Individual
The $80 Billion IT Spend

Percentages of Total IT Spending for Fiscal Years 2011 and 2012 for 26 Key Federal Agencies

Source: GAO based on OMB data.
The $592 Billion Opportunity

<table>
<thead>
<tr>
<th>Category</th>
<th>%</th>
<th>Spend</th>
<th>Waste</th>
<th>%</th>
<th>Annual Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td>30.0</td>
<td>24.0</td>
<td>Scrap and Rework</td>
<td>60.0</td>
<td>14.4</td>
</tr>
<tr>
<td>O &amp; M</td>
<td>70.0</td>
<td>56.0</td>
<td>Corrective Maintenance</td>
<td>80.0</td>
<td>44.8</td>
</tr>
<tr>
<td>Annual Spend</td>
<td>100.0</td>
<td>80.0</td>
<td></td>
<td></td>
<td>59.2</td>
</tr>
</tbody>
</table>
How Much Will FITARA Save?

• Management Of Information Technology Within Federal Government
• Data Center Optimization
• Elimination Of Duplication And Waste In Information Technology Acquisition
• Strengthening And Streamlining Information Technology Acquisition Management Practices
• Additional Reforms
BRIEFING ON ADDITIONAL MEANS OF IMPROVING SOFTWARE ASSURANCE.—Not later than one year after the date of the enactment of this Act, the Under Secretary for Acquisition, Technology, and Logistics shall, in coordination with the Chief Information Officer of the Department of Defense, provide to the congressional defense committees a briefing on the following:

(1) A research and development strategy to advance capabilities in software assurance and vulnerability detection.

(2) The state-of-the-art of software assurance analysis and test.

(3) How the Department might hold contractors liable for software defects or vulnerabilities.
The Way Forward

Seven

“Outrageous Commitments”
1. Create constancy of purpose and dedication to address the persistent wicked problems
Way Forward
“Outrageous Commitment” No. 1

Create constancy of purpose and dedication to address the persistent wicked problems

Establish five-year “to be” state and describe methods to achieve goals by connecting the dots: agility, quality, innovation and joy in work

Eliminate sub-optimal “deliver now, fix later” solutions
2. Restore the individual
Way Forward
“Outrageous Commitment” No. 2

Restore the individual

Establish minimum competency in planning and tracking the work and managing and measuring software quality

Abolish performance review and substitute with self evaluation and feedback with coaching support

Provide line of sight and explicit cause and effect linking individual performance to organization’s aim and team goals
3. End the practice of imposing arbitrary and unrealistic schedules
Way Forward

“Outrageous Commitment” No. 3

End the practice of imposing arbitrary and unrealistic schedules

Abolish arbitrary no. of sprints and time-bound releases

Ensure engineers have the skill to make a plan and the conviction to defend it

When pressed for early deliveries, the responsible team members say “I understand your requirements, I will do my utmost to meet it, but until I make a plan, I can not responsibly commit to a date”

Trust and support the teams

When pressed for early deliveries, the responsible managers say “I trust you to create an aggressive and realistic plan, I will review the plan, but I will not commit you to a date that you can not meet”
4. Improve continuously and forever the performance metrics that matter
Way Forward
“Outrageous Commitment” No. 4

Improve continuously and forever the performance metrics that matter

Cost and schedule performance, defect density in delivered software, percent of software reuse, phase yields, cost of quality, and process improvement proposals
5. Cease dependence on test and rework cycle for defect removal
Way Forward

“Outrageous Commitment” No. 5

Cease dependence on test and rework cycle for defect removal

Promote the practice of putting highest quality product into test

Strive for 100 percent defect free components in integration, system and acceptance testing

Improve capability to predict quality early in the software development lifecycle
6. Transform acquisition practices with less emphasis on oversight and create a culture of continuous improvement based on facts
Way Forward

“Outrageous Commitment” No. 6

6. Transform acquisition practices with less emphasis on oversight and create a culture of continuous improvement based on facts

Eliminate the practice of using maturity levels as pass/fail criteria

Substitute appraisals and certifications with measuring organizational competency in performance metrics that matter

Substitute bureaucratic status reporting with precise and accurate reporting of performance metrics that matter
7. Take action to fulfill the outrageous commitments
Way Forward
“Outrageous Commitment” No. 7

Take action to fulfill the outrageous commitments

Sponsor national dialogue on persistent wicked software engineering problems

Elaborate and refine the seven outrageous commitments

Leverage the purchasing power of the stakeholders to change acquisition as well as industry/supplier practices
Reactionary Arguments Against Solutions To Wicked Problems

Three basic forms: perversity, futility, and jeopardy

Perversity thesis: change will not just fail but make the problem worse

Futility thesis: change can’t make a meaningful difference, and therefore won’t be worth the effort

Jeopardy thesis: change will impose unacceptable costs upon society

Redundancy thesis: we are already doing it

http://www.newyorker.com/online/blogs/comment/2012/06/something-wicked-this-way-comes.html#ixzz25Lb6mG32
What If?

The government holds contractors accountable for cybersecurity vulnerabilities caused by defective software and require that contractors provide warranty against software defects?

The government procurement is based on Lowest Price Highest Quality (LPHQ) rather than Lowest Price Technically Acceptable (LPTA)?

We call software bugs, “bombs”?

We call technical debt, “malpractice”?  

We call test and debug “scrap and rework”?  

The government never mentions CMMI, SCRUM, PMP in solicitations?  

We eliminate two-week sprints?
Joy in Work

“There is a square; there is an oblong. The players take the square and place it upon the oblong. They place it very accurately; they make a perfect dwelling place. Very little is left outside. The structure is now visible; what was inchoate is here stated; we are not so various or so mean; we have made oblongs and stood them upon squares. This is our triumph; this is our consolation.”

The players in Virginia Woolf’s *The Waves*
What does “FUN ON THE JOB” Mean to you?
Girish Seshagiri

girish.seshagiri@advinfo.net

703 426-2790
Backup Slides
Performance Metrics That Matter
The Balanced Scorecard

Customer
“To achieve our vision, how should we appear to our customers?”

Financial
“To succeed financially, how should we appear to our shareholders?”

Vision and Strategy

Learning and Growth
“To achieve our vision, how will we sustain our ability to change and improve?”

Internal Business Processes
“To satisfy our shareholders and customers, what business processes must we excel at?”

Copyright AIS 2012, 2013
## Performance Metrics That Matter

### Leading, Lagging Indicators

<table>
<thead>
<tr>
<th>FINANCIAL</th>
<th>CUSTOMER</th>
<th>EMPLOYEE</th>
<th>INTERNAL BUSINESS PROCESS</th>
<th>LEARNING &amp; GROWTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Quarterly revenue increase&lt;br&gt;• Annual valuation increase</td>
<td>• Customer feedback indicating met or exceeded customer expectations for:&lt;br&gt;- Quality&lt;br&gt;- Timeliness&lt;br&gt;- Value for Products &amp; Services</td>
<td>• Targeted employee turnover rate</td>
<td>• Work products with less than or equal to targeted rework effort&lt;br&gt;• Work products with zero post development defects&lt;br&gt;• Commitments with actual effort equal to or less than committed effort&lt;br&gt;• Commitments with actual schedule equal to or less than committed schedule</td>
<td>• Teams achieving quantitative goals&lt;br&gt;• Employees advancing to the next level in the career path&lt;br&gt;• New products developed and released&lt;br&gt;• New services offered</td>
</tr>
</tbody>
</table>

### Core Outcomes – Lagging Indicators
- Monthly percent of revenue from targeted customers
- Monthly percent of revenue from targeted services
- Monthly percent of employees meeting or exceeding their budgeted revenue target
- Teams using the TSP
- Individuals following the PSP
- Work products with targeted percent of defects removed before peer review
- Teams using the TSP
- Employees achieving training goals
- Internal products being developed
- Identified market needs

## Performance Drivers – Leading Indicators
- Defect free delivery
- On-time delivery
- Assessment indicating targeted P-CMM Key Process Areas fully satisfied
- Individuals following the PSP
- Work products with targeted percent of defects removed before peer review
- Teams using the TSP
- Employees achieving training goals
- Internal products being developed
- Identified market needs

---

Copyright AIS 2012, 2013
Performance Metrics That Matter
Leading, Lagging Indicators - 2

<table>
<thead>
<tr>
<th>FINANCIAL</th>
<th>CUSTOMER</th>
<th>EMPLOYEE</th>
<th>INTERNAL BUSINESS PROCESS</th>
<th>LEARNING &amp; GROWTH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CORE OUTCOMES – LAGGING INDICATORS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quarterly revenue increase</td>
<td>• Customer feedback indicating met or exceeded customer expectations for:</td>
<td>• Targeted employee turnover rate</td>
<td>• Work products with less than or equal to targeted rework effort</td>
<td>• Teams achieving quantitative goals</td>
</tr>
<tr>
<td>• Annual valuation increase</td>
<td>- Quality</td>
<td></td>
<td>• Work products with zero post development defects</td>
<td>• Employees advancing to the next level in the career path</td>
</tr>
<tr>
<td></td>
<td>- Timeliness</td>
<td></td>
<td>• Commitments with actual effort equal to or less than committed effort</td>
<td>• New products developed and released</td>
</tr>
<tr>
<td></td>
<td>- Value for Products &amp; Services</td>
<td></td>
<td>• Commitments with actual schedule equal to or less than committed schedule</td>
<td>• New services offered</td>
</tr>
</tbody>
</table>

| PERFORMANCE DRIVERS – LEADING INDICATORS | | | | |
| • Monthly percent of revenue from targeted customers | • Defect free delivery | • Assessment indicating targeted P-CMM Key Process Areas fully satisfied | • Individuals following the PSP |
| • Monthly percent of revenue from targeted services | • On-time delivery | | • Work products with targeted percent of defects removed before peer review |
| • Monthly percent of employees meeting or exceeding their budgeted revenue target | | | | • Teams using the TSP |

Copyright AIS 2012, 2013
## Task Status Summary

<table>
<thead>
<tr>
<th>Status Summary</th>
<th>Plan</th>
<th>Actual</th>
<th>Pln/Act</th>
<th>Act/Pln</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Hours</td>
<td>450.3</td>
<td>468.3</td>
<td>0.96</td>
<td>1.04</td>
</tr>
<tr>
<td>Project Hours To-date</td>
<td>3173.0</td>
<td>3255.3</td>
<td>0.97</td>
<td>1.03</td>
</tr>
<tr>
<td>Earned Value</td>
<td>4.3</td>
<td>1.7</td>
<td>2.51</td>
<td>0.40</td>
</tr>
<tr>
<td>EV To-date</td>
<td>28.5</td>
<td>26.9</td>
<td>1.06</td>
<td>0.94</td>
</tr>
<tr>
<td>To-date Hours for EV Tasks Closed</td>
<td>2482.0</td>
<td>1977.1</td>
<td>1.26</td>
<td>0.80</td>
</tr>
<tr>
<td>To-date Hours for Non-EV Tasks</td>
<td>429.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Current Week – 7 of 29

<table>
<thead>
<tr>
<th>Current Status Based On EV</th>
<th>Week(s)</th>
<th>Projected Completion Based On</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Status Based On EV</td>
<td>0.4</td>
<td>Behind</td>
</tr>
<tr>
<td>Projected Completion Based On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Earned Value Earn Rate</td>
<td>3.0</td>
<td>Ahead</td>
</tr>
<tr>
<td>Remaining Plan Hours and EV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Hours and Estimate Accuracy</td>
<td>4.9</td>
<td>Ahead</td>
</tr>
</tbody>
</table>

| Blocked EV Effort | 848.6 |
| Avg EV Hours Per Week | 465.0 |
| Avg EV Per Week | 3.8 |
| EV Hours req'd for 100% EV | 5219.9 |
| For ontime completion | Per Week |
| Required EV | 3.3 |
| Required EV Hours | 237.3 |
### Task Status Summary

<table>
<thead>
<tr>
<th>Status Summary</th>
<th>Plan</th>
<th>Actual</th>
<th>Pln/Act</th>
<th>Act/Pln</th>
<th>Current Status Based On EV</th>
<th>7 of 29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Hours</td>
<td>38.0</td>
<td>41.1</td>
<td>0.92</td>
<td>1.08</td>
<td></td>
<td>Week(s)</td>
</tr>
<tr>
<td>Project Hours To-date</td>
<td>269.7</td>
<td>268.8</td>
<td>0.99</td>
<td>1.11</td>
<td></td>
<td>4.8</td>
</tr>
<tr>
<td>Earned Value</td>
<td>8.1</td>
<td>6.7</td>
<td>1.22</td>
<td>0.82</td>
<td></td>
<td>Behind</td>
</tr>
<tr>
<td>EV To-date</td>
<td>62.6</td>
<td>36.9</td>
<td>1.70</td>
<td>0.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To-date Hours for EV Tasks Close</td>
<td>141.9</td>
<td>269.4</td>
<td>0.53</td>
<td>1.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To-date Hours for Non-EV Tasks</td>
<td>20.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Current Week

<table>
<thead>
<tr>
<th>Current Status Based On EV</th>
<th>Week(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 of 29</td>
<td>4.8</td>
</tr>
</tbody>
</table>

### Projected Completion Based On

<table>
<thead>
<tr>
<th>Earned Value Earn Rate</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remaining Planned Hours and EV</td>
<td></td>
</tr>
<tr>
<td>Project Hours and Estimate Accuracy</td>
<td>15.7</td>
</tr>
</tbody>
</table>

### Blocked EV Effort

<table>
<thead>
<tr>
<th>Blocked EV Effort</th>
<th>8.7</th>
</tr>
</thead>
</table>

| Avg EV Hours Per Week | 42.7 |
| Avg EV Per Week       | 5.3  |
| EV Hours reqd for 100% EV | 460.4 |

<table>
<thead>
<tr>
<th>For ontime completion</th>
<th>Per Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required EV</td>
<td>2.9</td>
</tr>
<tr>
<td>Required EV Hours</td>
<td>20.9</td>
</tr>
</tbody>
</table>

---

Performance Metrics That Matter

Quality Profile

The above chart shows our 5 measures to achieve our quality goals:

1) Design time to Code time comparison
2) Design Review Time as a % of design time (should be 50% or greater)
3) Code Review Time as a % of code time (should be 50% or greater)
4) Compile Defects per KLOC (Compile should find less than 10 defects per KLOC)
5) Unit Test Defects per KLOC (UT should find less than 5 defects per KLOC)
Performance Metrics That Matter
Defects Removed by Phase

Defects Removed by Phase for Assembly SYSTEM
Performance Metrics That Matter
Process Improvement Proposal Activity

From October 1992 to August 2012

1741 PIPs have been submitted

Approximately 1255 PIPs have been implemented
Using Organization Data for Project Planning - Regression

\[ y = 1.1348x + 71.685 \]

\[ R^2 = 0.9764 \]
Performance Metrics That Matter

Post-delivery Defects - 1

User Acceptance Test Defects Per KLOC - New Development

(Avg=0.1606, UCL=0.6643)
Performance Metrics That Matter
Post-delivery Defects - 2

User Acceptance Test Defects Per KLOC - Enhancement
(Avg=0.3765, UCL=1.3867)
Continuous Improvement

Data is captured and analyzed after each development cycle
Estimated and actual size, effort and defects builds personal history
Regression analysis continuously improves estimation
Analysis of defects results in actions to prevent similar defects in the future