Future Systems and Software Challenges, Especially Maintainability

Barry Boehm, USC
USC-CSSE Annual Research Review
April 4, 2017
Future systems and software engineering challenges

- Humans and autonomy, systems of systems, internets of things, 3D printing, cyber security, cloud computing, legacy maintainability
- Example: Software development cost estimation and COCOMO III

Maintainability shortfalls impact all aspects of software-intensive system (SIS) cost-effectiveness

- System life-cycle cost, dependability, changeability, mission effectiveness, resilience
- Increasing costs of software maintenance and technical debt

Both automated and human maintainability and technical debt evaluation methods effective and complementary

Root causes of technical debt primarily non-technical

- Root cause monitoring: SIS Maintenance Readiness Framework (SMRF)

Conclusions
A Short History of Software Estimation Accuracy

IDPD: Incremental Development Productivity Decline
MBSE: Model-Based Systems and Sw Engr.
COTS: Commercial Off-the-Shelf
SoS: Systems of Systems

Relative Productivity
Estimation Error

Unprecedented
Precedent
Component-based
COTS
Agile
SoS. Apps, Widgets, IDPD, Clouds, Security, MBSE

Time, Domain Understanding
COCOMO Family of Cost Models

Software Cost Models

- COCOMO 81 1981
- COCOMO II 2000
- DBA COCOMO 2004
- COINCOMO 2004, 2012
- COCOTS 2000
- COSYSIMO 2005
- COSoSIMO 2007
- CORADMO 1999, 2012

Software Extensions

- COQUALMO 1998
- iDAVE 2004
- COPLIMO 2003
- COPSEMO 1998
- AGILE C II 2003
- COTIPMO 2011
- COPROMO 1998
- COSECIMO 2004

Legend:
- Model has been calibrated with historical project data and expert (Delphi) data
- Model is derived from COCOMO II
- Model has been calibrated with expert (Delphi) data

Dates indicate the time that the first paper was published for the model
Estimation-Type Options

- Expert-Judgement; Stakeholder Consensus
  - Planning Poker, Wideband Delphi, Bottom-Up
- Analogy: Previous Projects; Yesterday’s Weather
  - Agile COCOMO II, Case-Based Reasoning, IDPD
- Parametric Models
  - COCOMO/COSTAR, Knowledge Plan, SEER, SLIM, True-S
- Resource-Limited
  - Cost or Schedule as Independent Variable (CAIV, SAIV)
- Reuse-Driven: Equivalent Size
  - Adjusted for %Design, Code, Test Modified, Understandability
- Product Line
  - % Development for Reuse; % Development with Reuse
• Pure Agile: Planning Poker, Agile COCOMO II

• Architected Agile
  – COSYSMO for architecting; Planning Poker, CAIV-SAIV for sprints, releases; IDPD for large systems

• Formal Methods: $/SLOC by Evaluated Assurance Level

• NDI/Services-Intensive: Oracle, SAP, other ERP
  – RICE Objects: (R)eports, (I)nterfaces, (C)onversions, (E)nhancements
  – COCOTS, Value-Added Function Points, Agile for portions

• Hybrid Agile/Plan-Driven
  – Expert Delphi, Parametric Models, Agile for portions; IDPD

• Systems of Systems
  – COSYSMO for Integrator; Hybrid Agile/Plan-Driven for component systems

• Family of Systems: COPLIMO

• Brownfield: Experiment for refactoring; above for rebuilding
Persistence of Legacy Systems

- Before establishing new-system increments
  - Determine how to undo legacy system

1939’s Science Fiction World of 2000

Actual World of 2000
Outline

• Future systems and software engineering challenges
  – Humans and autonomy, systems of systems, internets of things, 3D printing, cyber security, cloud computing, legacy maintainability
  – Example: Software development cost estimation and COCOMO III

Maintainability shortfalls impact all aspects of software-intensive system (SIS) cost-effectiveness
  – System life-cycle cost, dependability, changeability, mission effectiveness, resilience
  – Increasing costs of software maintenance and technical debt

• Both automated and human maintainability and technical debt evaluation methods effective and complementary

• Root causes of technical debt primarily non-technical
  – Root cause monitoring: SIS Maintenance Readiness Framework (SMRF)

• Conclusions
Dependability, Changeability, and Resilience

- **Reliability**
  - Defect Freedom
  - Survivability
  - Fault Tolerance
  - Complete
  - Partial
    - Robustness
    - Self-Repairability
    - Graceful Degradation

- **Dependability, Availability**
- **Maintainability**
- **Repairability**
- **Testability**
- **Resilience**
- **Changeability**
- **Modifiability**

**Test Plans, Coverage**
**Test Scenarios, Data**
**Test Drivers, Oracles**
**Test Software Qualities**

Means to End
Subclass of

Testability, Diagnosability, etc.
Cost of Downtime Survey

- **Industry Sector Revenue/Hour**
- **Energy** $2.8 million
- **Telecommunications** $2.0 million
- **Manufacturing** $1.6 million
- **Financial Institutions** $1.4 million
- **Information Technology** $1.3 million
- **Insurance** $1.2 million
- **Retail** $1.1 million
- **Pharmaceuticals** $1.0 million
- **Banking** $996,000

• Future systems and software engineering challenges
  – Humans and autonomy, systems of systems, internets of things, 3D printing, cyber security, cloud computing, legacy maintainability
  – Example: Software development cost estimation and COCOMO III

• Maintainability shortfalls impact all aspects of software-intensive system (SIS) cost-effectiveness
  – System life-cycle cost, dependability, changeability, mission effectiveness, resilience
  – Increasing costs of software maintenance and technical debt

Both automated and human maintainability and technical debt evaluation methods effective and complementary

• Root causes of technical debt primarily non-technical
  – Root cause monitoring: SIS Maintenance Readiness Framework (SMRF)

• Conclusions
Software Maintainability Analysis Options

- Automated code analysis:
  - Maintainability Index
  - Technical Debt

- Human-based analysis:
  - Reuse cost models that estimate maintainability of potentially reusable components based on human-assessed aspects such as code understandability and structure
    - COCOMO II Software Understandability Factors
Maintainability Index

● “The ease in which a system can be modified or extended”
● Maintainability Index (MI)
  ○ An index that represents the ease of maintaining the code
  ○ Widely used in the industry
Maintainability Index

\[
MI_{woc(sourcefile)} = 171 - 5.2 \times \ln HV - 0.23 \times CC - 16.2 \times \ln LLOC
\]

\[
MI_{wc(sourcefile)} = 50 \times \sin \sqrt{2.46 \times CM}
\]

\[
MI_{(sourcefile)} = MI_{woc(sourcefile)} + MI_{wc(sourcefile)}
\]

\[
MI = \frac{\sum MI_{(sourcefile)}}{Number \ of \ Source \ files}
\]

Halstead Volume (HV)  Cyclomatic complexity (CC)
Count of lines (LLOC)  Percent of lines of comments (CM)

\textit{MI is developed by the University of Idaho in 1991 by Oman and Hagemeister}
What is Technical Debt (TD)?

- TD: Delayed technical work or rework that is incurred when short-cuts are taken or short-term needs are addressed first
  - The later you pay for it, the more it costs (interest on debt)
- Global Information Technology Technical Debt [Gartner 2010]
  - 2010: Over $500 Billion; By 2015: Over $1 Trillion
- TD as Investment
  - Competing for first-to-market
  - Risk assessment: Build-upon prototype of key elements
  - Rapid fielding of defenses from terrorist threats
- TD as Lack of Foresight
  - Overfocus on Development vs. Life Cycle
  - Skimping on Systems Engineering
  - Hyper-Agile Development: Easiest-First increments
    - Neglecting Rainy-Day Use Cases, Non-Functional Requirements
Example of Technical Debt Calculation (SonarQube)

Debt (in man days) = cost_to_fix_duplications 
+ cost_to_fix_violations 
+ cost_to_comment_public_API 
+ cost_to_fix_uncovered_complexity 
+ cost_toBring_complexity_below_threshold
Pros & Cons of Automated Maintainability Metrics

• Pros:
  - Very popular and often used in maintenance practice
  - Very easy to use

• Cons:
  - Most automated metrics are composite metrics and as such it is hard to determine which of the metrics cause a particular total value
  - The different metric values depend on the type of programming language, the programmer, the perception of the quality of code, etc.
<table>
<thead>
<tr>
<th>Factor</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Clarity</td>
<td>No Match between program an application worldviews.</td>
<td>Some correlation between program and application.</td>
<td>Moderate correlation between program and application.</td>
<td>Good correlation between program and application.</td>
<td>Clear match between program and application worldviews.</td>
</tr>
<tr>
<td>Self-Descriptiveness</td>
<td>Obscure code; documentation missing, obscure or obsolete.</td>
<td>Some code commentary and headers; some useful documentation.</td>
<td>Moderate level of code commentary, headers, documentation.</td>
<td>Good code commentary and headers; useful documentation; some weak areas.</td>
<td>Self-descriptive code; documentation up-to-date, well organized, with design rationale.</td>
</tr>
</tbody>
</table>
Controlled Experiment Setup

- **Tasks**: maintenance tasks
  - Students performed maintenance tasks (bug fixing or new feature requests implementation) on 11 projects.
  - Overall 44 tasks were completed – 28 bug fixing tasks and 16 new feature tasks.

<table>
<thead>
<tr>
<th>Student</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Projects</td>
<td>11</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>Number of Finished Tasks</td>
<td>35</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>13</td>
<td>31</td>
</tr>
<tr>
<td>Number of Unfinished Tasks</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>9</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>Number of Total Tasks</td>
<td>44</td>
<td>20</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>44</td>
</tr>
<tr>
<td>% of Finished Tasks</td>
<td>79.55</td>
<td>60.00</td>
<td>75.00</td>
<td>43.75</td>
<td>65.00</td>
<td>70.45</td>
</tr>
</tbody>
</table>
Characteristics of Project Data Sources

Projects were selected from Apache community & SourceForge.

Java Projects:
  • Docfetcher, DrJava, Aerosolve, Pinot, Auil, PMD

PHP Projects:
  • Joomla, Composer, Codeception, Pagekit, PhpUnit
Correlation between SU factors and maintenance effort

- All three COCOMO II SU Factors showed strong negative relationship with maintenance effort.
  - Higher quality structure was correlated with less effort spent on maintenance tasks.
  - When software reflects higher application content clarity, developers spent less effort on maintenance tasks.
  - More self-descriptive source code was correlated with less effort spent on maintenance tasks.

- However, we didn’t find any relationship between documentation quality and maintenance effort.
Comparison and Conclusions

- The human-assessed maintainability metrics were more accurate in estimating software maintainability effort.

- The automated maintainability metrics were reasonably accurate in estimating software maintainability effort.
  - But were much better in determining which parts of the software were the primary sources of added maintenance effort; particularly for technical debt.
  - And much more efficient than human assessment.

- Neither human or automated approaches dominate.
  - Particularly for large projects, best to have automated methods determine the main sources of added maintenance effort, and use human assessment on the critical parts.
• Future systems and software engineering challenges
  – Humans and autonomy, systems of systems, internets of things, 3D printing, cyber security, cloud computing, legacy maintainability
  – Example: Software development cost estimation and COCOMO III

• Maintainability shortfalls impact all aspects of software-intensive system (SIS) cost-effectiveness
  – System life-cycle cost, dependability, changeability, mission effectiveness, resilience
  – Increasing costs of software maintenance and technical debt

• Both automated and human maintainability and technical debt evaluation methods effective and complementary

Root causes of technical debt (TD) primarily non-technical
  – Root cause monitoring: SIS Maintenance Readiness Framework (SMRF)

• Conclusions
Top-10 Non-Technical Sources of TD

1. Separate organizations and budgets for systems and software acquisition and maintenance
2. Overconcern with the Voice of the Customer
3. The Conspiracy of Optimism
4. Inadequate system engineering resources
5. Hasty contracting that focuses on fixed operational requirements
6. CAIV-limited system requirements
7. Brittle, point-solution architectures
8. The Vicious Circle
9. Stovepipe systems
10. Over-extreme forms of agile development
1. Separate R&D, O&M Organizations, Budgets
Problem and Opportunity (%O&M costs)

- **US Government IT:** >73%; $58 Billion [GAO 2015]
- **Cyber-Physical Systems** [Redman 2008]
  - 12% -- Missiles (average)
  - 60% -- Ships (average)
  - 78% -- Aircraft (F-16)
    - 1960: 8% of functionality in software; 2000: 80% [Ferguson 2001]
    - 84% -- Ground vehicles (Bradley)
- **Software** [Koskinen 2010]
  - 75-90% -- Business, Command-Control
  - 50-80% -- Complex platforms as above
  - 10-30% -- Simple embedded software
- **Primary current emphases** minimize acquisition costs
2. Overconcern with the Voice of the Customer/User

Bank of America Master Net
3. The Conspiracy of Optimism

Take the lower branch of the Cone of Uncertainty

---

**Aerospace America, 1/2016**

- **F-22**
  - 187 A/C
  - $79B

- **F-22**
  - 750 A/C
  - $26B
4. Inadequate system engineering resources

Sweet Spot Drivers:
- Rapid Change: leftward
- High Assurance: rightward
5. Hasty contracting that focuses on fixed operational requirements
   5. Fixed price contract to minimum-cost, technically-acceptable bidder

6. CAIV-limited system requirements
   5. Below-threshold capabilities dropped from RFP, losing evolution insight

7. Brittle, point-solution architectures
   5. Result of 5,6. Need set-based design

8. The Vicious Circle
   5. Maintainers: We wish we could participate in the acquisition process, but we’re overloaded in fixing TD problems from similar previous acquisitions

9. Stovepipe systems
   5. Sharing common elements means they are upgraded once vs. N times

10. Over-extreme forms of agile development
    5. Easiest-first increments; Sunny-day use cases; Defer quality requirements
### Software-Intensive Systems Maintainability Readiness Levels

<table>
<thead>
<tr>
<th>SMR Level</th>
<th>OpCon, Contracting: Missions, Scenarios, Resources, Incentives</th>
<th>Personnel Capabilities and Participation</th>
<th>Enabling Methods, Processes, and Tools (MPTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>5 years of successful maintenance operations, including outcome-based incentives, adaptation to new technologies, missions, and stakeholders</td>
<td>In addition, creating incentives for continuing effective maintainability, performance on long-duration projects</td>
<td>Evidence of improvements in innovative O&amp;M MPTs based on ongoing O&amp;M experience</td>
</tr>
<tr>
<td>8</td>
<td>One year of successful maintenance operations, including outcome-based incentives, refinements of OpCon.</td>
<td>Stimulating and applying People CMM Level 5 maintainability practices in continuous improvement and innovation in such technology areas as smart systems, use of multicore processors, and 3-D printing</td>
<td>Evidence of MPT improvements based on ongoing refinement, and extensions of ongoing evaluation, initial O&amp;M MPTs.</td>
</tr>
<tr>
<td>7</td>
<td>System passes Maintainability Readiness Review with evidence of viable OpCon, Contracting, Logistics, Resources, Incentives, personnel capabilities, enabling MPTs</td>
<td>Achieving advanced People CMM Level 4 maintainability capabilities such as empowered work groups, mentoring, quantitative performance management and competency-based assets, particularly across key domains.</td>
<td>Advanced, integrated, tested, and exercised full-LC MBS&amp;SE MPTs and Maintainability-other-SQ tradespace analysis</td>
</tr>
<tr>
<td>6</td>
<td>Mostly-elaborated maintainability OpCon, with roles, responsibilities, workflows, logistics management plans with budgets, schedules, resources, staffing, infrastructure and enabling MPT choices, V&amp;V and review procedures.</td>
<td>Achieving basic People CMM levels 2 and 3 maintainability practices such as maintainability work environment, competency and career development, and performance management especially in such key areas such as V&amp;V, identification &amp; reduction of technical debt.</td>
<td>Advanced, integrated, tested full-LC Model-Based Software &amp; Systems (MBS&amp;SE) MPTs and Maintainability-other-SQ tradespace analysis tools identified for use, and being individually used and integrated.</td>
</tr>
<tr>
<td>5</td>
<td>Convergence, involvement of main maintainability success-critical stakeholders. Some maintainability use cases defined. Rough maintainability OpCon, other success-critical stakeholders, staffing, resource estimates. Preparation for NDI and outsource selections.</td>
<td>In addition, independent maintainability experts participate in project evidence-based decision reviews, identify potential maintainability conflicts with other SQs</td>
<td>Advanced full-lifecycle (full-LC) O&amp;M MPTs and SW/SE MPTs identified for use. Basic MPTs for tradespace analysis among maintainability &amp; other SQs, including TCO being used.</td>
</tr>
<tr>
<td>4</td>
<td>Artifacts focused on missions. Primary maintenance options determined, Early involvement of maintainability success-critical stakeholders in elaborating and evaluating maintenance options.</td>
<td>Critical mass of maintainability SysEs with mission SysE capability, coverage of full M-SysE.skills areas, representation of maintainability success-critical-stakeholder organizations.</td>
<td>Advanced O&amp;M MPT capabilities identified for use: Model-Based SW/SE, TCO analysis support. Basic O&amp;M MPT capabilities for modification, repair and V&amp;V: some initial use.</td>
</tr>
<tr>
<td>2</td>
<td>Mission evolution directions and maintainability implications explored. Some mission use cases defined, some O&amp;M options explored.</td>
<td>Highly maintainability-capable SysEs included in Early SysE team.</td>
<td>Initial exploration of O&amp;M MPT options</td>
</tr>
<tr>
<td>1</td>
<td>Focus on mission opportunities, needs. Maintainability not yet considered</td>
<td>Awareness of needs for early expertise for maintainability. concurrent engr’g, O&amp;M integration, Life Cycle cost estimation</td>
<td>Focus on O&amp;M MPT options considered</td>
</tr>
</tbody>
</table>

---

**SIS Maintainability Readiness Framework (SMRF)**

---

**6/10/2016 31**
## Software-Intensive Systems Maintainability Readiness Levels

<table>
<thead>
<tr>
<th>SMR Level</th>
<th>OpCon, Contracting: Missions, Scenarios, Resources, Incentives</th>
<th>Personnel Capabilities and Participation</th>
<th>Enabling Methods, Processes, and Tools (MPTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>7</strong></td>
<td>System passes Maintainability Readiness Review with evidence of viable OpCon, Contracting, Logistics, Resources, Incentives, personnel capabilities, enabling MPTs</td>
<td>Achieving advanced People CMM Level 4 maintainability capabilities such as empowered work groups, mentoring, quantitative performance management and competency-based assets, particularly across key domains.</td>
<td>Advanced, integrated, tested, and exercised full-LC MBS&amp;SE MPTs and Maintainability-other-SQ tradespace analysis tools identified for use, and being individually used and integrated.</td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>Mostly-elaborated maintainability OpCon. with roles, responsibilities, workflows, logistics management plans with budgets, schedules, resources, staffing, infrastructure and enabling MPT choices, V&amp;V and review procedures.</td>
<td>Achieving basic People CMM levels 2 and 3 maintainability practices such as maintainability work environment, competency and career development, and performance management especially in such key areas such as V&amp;V, identification &amp; reduction of technical debt.</td>
<td>Advanced, integrated, tested full-LC Model-Based Software &amp; Systems (MBS&amp;SE) MPTs and Maintainability-other-SQ tradespace analysis tools identified for use, and being individually used and integrated.</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>Convergence, involvement of main maintainability success-critical stakeholders. Some maintainability use cases defined. Rough maintainability OpCon, other success-critical stakeholders, staffing, resource estimates. Preparation for NDI and outsource selections.</td>
<td>In addition, independent maintainability experts participate in project evidence-based decision reviews, identify potential maintainability conflicts with other SQs</td>
<td>Advanced full-lifecycle (full-LC) O&amp;M MPTs and SW/SE MPTs identified for use. Basic MPTs for tradespace analysis among maintainability &amp; other SQs, including TCO being used.</td>
</tr>
</tbody>
</table>
Conclusions

• Many future systems and software engineering challenges
  – Humans and autonomy, systems of systems, internets of things, 3D printing, cyber security, cloud computing, legacy maintainability
• Maintainability shortfalls impact all aspects of system cost-effectiveness
  – System life-cycle cost, dependability, changeability, mission effectiveness, resilience
  – Increasing costs of software maintenance and technical debt
  – Useful human and automated maintainability analysis tools
• Root causes explain sources of software-intensive systems (SIS) life cycle cost escalation
• SIS Maintenance Readiness Framework (SMRF) enables projects to confront and overcome the root causes
### SIS Maintainability Readiness Levels 1-3

Software-Intensive Systems Maintainability Readiness Levels

<table>
<thead>
<tr>
<th>SMR Level</th>
<th>OpCon, Contracting: Missions, Scenarios, Resources, Incentives</th>
<th>Personnel Capabilities and Participation</th>
<th>Enabling Methods, Processes, and Tools (MPTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Mission evolution directions and maintainability implications explored. Some mission use cases defined, some O&amp;M options explored.</td>
<td>Highly maintainability-capable SysEs included in Early SysE team.</td>
<td>Initial exploration of O&amp;M MPT options</td>
</tr>
<tr>
<td>1</td>
<td>Focus on mission opportunities, needs. Maintainability not yet considered</td>
<td>Awareness of needs for early expertise for maintainability. concurrent engr'g, O&amp;M integration, Life Cycle cost estimation</td>
<td>Focus on O&amp;M MPT options considered</td>
</tr>
<tr>
<td>SMR Level</td>
<td>OpCon, Contracting: Missions, Scenarios, Resources, Incentives</td>
<td>Personnel Capabilities and Participation</td>
<td>Enabling Methods, Processes, and Tools (MPTs)</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>5</td>
<td>Convergence, involvement of main maintainability success-critical stakeholders. Some maintainability use cases defined. Rough maintainability OpCon, other success-critical stakeholders, staffing, resource estimates. Preparation for NDI and outsource selections.</td>
<td>In addition, independent maintainability experts participate in project evidence-based decision reviews, identify potential maintainability conflicts with other SQs</td>
<td>Advanced full-lifecycle (full-LC) O&amp;M MPTs and SW/SE MPTs identified for use. Basic MPTs for tradespace analysis among maintainability &amp; other SQs, including TCO being used.</td>
</tr>
<tr>
<td>4</td>
<td>Artifacts focused on missions. Primary maintenance options determined, Early involvement of maintainability success-critical stakeholders in elaborating and evaluating maintenance options.</td>
<td>Critical mass of maintainability SysEs with mission SysE capability, coverage of full M-SysE.skills areas, representation of maintainability success-critical-stakeholder organizations.</td>
<td>Advanced O&amp;M MPT capabilities identified for use: Model-Based SW/SE, TCO analysis support. Basic O&amp;M MPT capabilities for modification, repair and V&amp;V: some initial use.</td>
</tr>
</tbody>
</table>
## Software-Intensive Systems Maintainability Readiness Levels

<table>
<thead>
<tr>
<th>SMR Level</th>
<th>OpCon, Contracting: Missions, Scenarios, Resources, Incentives</th>
<th>Personnel Capabilities and Participation</th>
<th>Enabling Methods, Processes, and Tools (MPTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>5 years of successful maintenance operations, including outcome-based incentives, adaptation to new technologies, missions, and stakeholders</td>
<td>In addition, creating incentives for continuing effective maintainability, performance on long-duration projects</td>
<td>Evidence of improvements in innovative O&amp;M MPTs based on ongoing O&amp;M experience</td>
</tr>
<tr>
<td>8</td>
<td>One year of successful maintenance operations, including outcome-based incentives, refinements of OpCon.</td>
<td>Stimulating and applying People CMM Level 5 maintainability practices in continuous improvement and innovation in such technology areas as smart systems, use of multicore processors, and 3-D printing</td>
<td>Evidence of MPT improvements based on ongoing refinement, and extensions of ongoing evaluation, initial O&amp;M MPTs.</td>
</tr>
<tr>
<td>7</td>
<td>System passes Maintainability Readiness Review with evidence of viable OpCon, Contracting, Logistics, Resources, Incentives, personnel capabilities, enabling MPTs</td>
<td>Achieving advanced People CMM Level 4 maintainability capabilities such as empowered work groups, mentoring, quantitative performance management and competency-based assets, particularly across key domains.</td>
<td>Advanced, integrated, tested, and exercised full-LC MBS&amp;SE MPTs and Maintainability-other-SQ tradespace analysis</td>
</tr>
<tr>
<td>SMRL Level Vs. DoD Milestone</td>
<td>Simple, Non-Critical, Organic</td>
<td>Simple, Non-Critical, Transitioned</td>
<td>Intermediate</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>MDD</td>
<td>1</td>
<td>1</td>
<td>2-3</td>
</tr>
<tr>
<td>MS A</td>
<td>2</td>
<td>3</td>
<td>4-5</td>
</tr>
<tr>
<td>MS B</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>IOC</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>