Applying Agility to DoD
Common Operating Platform Environment Initiatives

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of EECS
Vanderbilt University
Nashville, Tennessee

SEI Agile Research Forum, May 22nd, 2012
DoD Strategic Acquisition Goals

- Deliver *enhanced* integrated warfighting capability at *lower cost* over the lifecycle & across the enterprise

- Reduce acquisition & new technology insertion cycle time

- Establish *sustainable* business & workforce strategies to support these goals

Alleviating complexities of software is crucial to meeting DoD acquisition goals
The DoD cannot achieve its strategic acquisition goals when it must support too many software development activities, each implementing a unique solution.

**Drawbacks**

- Stove-piped DoD solutions are redundant, proprietary, & brittle
- Expensive to develop, integrate, certify, & sustain
- Vulnerable to exploits

- Other problems with stove-pipes include non-scalable tactical performance
- Inadequate quality-of-service (QoS) for common real-time operating picture & distributed weapons control
Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes
Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Mandating common & portable interfaces based on open standards
- Ensuring full disclosure of design specs to competitors & small businesses
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes
Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes

![Diagram of Common Operating Platform Environments (COPEs)]
Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes
Solution: Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes
Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities

Ensuring full disclosure of design specs to competitors & small businesses

Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.

Mandating common & portable interfaces based on open standards

Achieving interoperability between system hardware & software applications via common protocols & data models

Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes
Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes
COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels

Provide mechanisms to manage endsystem resources, e.g., CPU scheduling, memory management, file systems & IPC
COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels

Domain-independent commonality

Encapsulates & enhances native OS mechanisms to create reusable network programming components
COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels.

Common Middleware Services

Distribution Middleware

Host Infrastructure Middleware

Operating Systems & Protocols

Domain-independent commonality

Defines higher-level distributed programming models whose reusable APIs & components automate & extend native OS capabilities.
COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels

Domain-independent commonality

Augment distribution middleware by defining higher-level domain-independent services that focus on programming “business logic”
COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels

Domain-specific commonality

Tailored to the requirements of particular domains, such as C4ISR, air defense, anti-submarine warfare, mission computing, land attack, etc.
COPE Benefits

Reduce acquisition & new technology insertion cycle time
COPE Benefits

Helps establish sustainable business & workforce strategies to support DoD goals
What’s Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive.

Serialized phasing of COPE infrastructure & application development postpones identifying design flaws that degrade system QoS until late in the lifecycle, i.e., during system integration.
What’s Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive.

Glacial contracting processes don’t support timely delivery of COPE capabilities to meet mission needs.
What’s Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive.

Contracting models that assume COPE requirements can be fully defined up front are expensive when inevitable changes occur.
What’s Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive.

QoS suffers when COPE initiatives attempt to use COTS products that are not suited for mission-critical DoD combat systems.
Rigid adherence to ossified standards & reference architectures impedes COPE technology refresh & limits application capabilities

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive.
What’s Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive.

At the heart of these problems is the lack of an holistic approach that balances key business, managerial, & technical drivers at scale.
Key Success Drivers for COPE Initiatives

COPE initiatives need an agile multi-dimensional perspective to foster success

Management Drivers
Ensuring effective leadership & guidance of COPE initiatives

Business Drivers
Achieving effective governance & broad acceptance of COPE economic aspects

Technical Drivers
Foundations of COPE development

Agile Architecture Expertise
Systematic Reuse Expertise
Data Rights & Effective Licensing Model
Agile Contracting Model
Managed Industry/Government Consortia
Understand the Strategic Role of Software
Automation Conformance & Regression Test suites
Strong S&T Connections to Reduce Risk
Mastery of Agile Lifecycle Methods
Applying Agility to Manage COPE Lifecycle

- Enable close cooperation of users, developers, testers, & certifiers throughout lifecycle to rapidly deliver COPE capabilities and avoid integration “surprises” without needing extensive upfront planning & serialized phasing.

- Emphasize incremental rollout of COPEs by delivering useful capability every 4 to 8 months to reduce risk via early validation by application developers & users.

Management Drivers
Ensuring effective leadership & guidance of COPE initiatives

Mastery of Agile Lifecycle Methods

dodcio.defense.gov/Portals/0/Documents/ITMod/CIO%2010%20Point%20Plan%20for%20IT%20Modernization.pdf
Applying Agility to Expedite Contracting

- Engage users & testers in developing COPE contract scope, evaluation criteria, incentives, & terms/conditions to ensure contracting supports all needs/considerations

- Expedite execution of COPE work packages via multiple award Indefinite-Delivery, Indefinite-Quantity (IDIQ) contract vehicles, & issue Task/Delivery Orders for each release

Business Drivers
Achieving effective governance & broad acceptance of COPE economic aspects

dodcio.defense.gov/Portals/0/Documents/ITMod/CIO%2010%20Point%20Plan%20for%20IT%20Modernization.pdf
Applying Agility to Ensure Architectural Flexibility

- Leverage common development, test & production platforms, & QoS-enabled standards-based COTS to deliver COPE capabilities faster, cheaper, & more interoperably, without redundant \textit{ad hoc} infrastructure

- Establish a change-tolerant architecture enabled by discovery learning that promotes decisions based on empirical data/evidence, rather than forecasts or legacy commitments

Technical Drivers
Foundations of COPE development

Agile Architecture Expertise

cope

dodcio.defense.gov/Portals/0/Documents/ITMod/CIO%202010%20Point%20Plan%20for%20IT%20Modernization.pdf
<table>
<thead>
<tr>
<th>COPE Challenges</th>
<th>How Agility Helps Resolve COPE Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serialized phasing of COPE infrastructure &amp; application development postpones identifying design flaws that degrade system QoS until late in the lifecycle, i.e., during system integration</td>
<td>Enables close cooperation of users, developers, testers, &amp; certifiers throughout lifecycle to rapidly deliver COPE capabilities and avoid integration “surprises” without needing extensive upfront planning &amp; serialized phasing</td>
</tr>
<tr>
<td>Glacial contracting processes don’t support timely delivery of COPE capabilities to meet mission needs</td>
<td>Engages users &amp; testers in developing COPE contract scope, evaluation criteria, incentives, &amp; terms/conditions to ensure contracting supports all needs/considerations</td>
</tr>
<tr>
<td>Contracting models that assume COPE requirements can be defined fully up front are expensive when inevitable changes occur</td>
<td>Expedites execution of COPE work packages via multiple award Indefinite-Delivery, Indefinite-Quantity (IDIQ) contract vehicles, &amp; issue Task/Delivery Orders for each release</td>
</tr>
<tr>
<td>QoS suffers when COPE initiatives attempt to use COTS products that are not suited for mission-critical DoD combat systems</td>
<td>Leverages common development, test &amp; production platforms, &amp; QoS-enabled standards-based COTS to deliver COPE capabilities faster, cheaper, &amp; more interoperably, without redundant ad hoc infrastructure</td>
</tr>
<tr>
<td>Rigid adherence to ossified standards &amp; reference architectures impedes COPE technology refresh &amp; limits application capabilities</td>
<td>Establishes a change-tolerant architecture enabled by discovery learning that promotes decisions based on empirical data/evidence, rather than forecasts or legacy commitments</td>
</tr>
</tbody>
</table>
Concluding Remarks

- DoD COPE initiatives for defense systems need a holistic strategy

  - Strong S&T Connections to Reduce Risk
  - Mastery of Agile Lifecycle Methods
  - Understand the Strategic Role of Software
  - Agile Architecture Expertise
  - Automated Conformance & Regression Test suites
  - Systematic Reuse Expertise
  - Agile Contracting Models
  - Data Rights & Effective Licensing Model
  - Managed Industry/Government Consortia

- DoD COPEs are achievable & valuable, though not easy to develop & sustain
- Agility in business, management, & technical dimensions is essential, but no panacea

We need your help developing successful ecosystems that promote common processes, procedures, software services, & systematic reuse of capability across COPE layers