

# Implementing COTS Systems

Is Agility Even Possible?



# Agenda

- Why are we here?
- What are COTS Systems?
  - General terms
  - Common examples you may already use
- Why COTS Systems need to be customized
- A reminder of our Agile roots and what Agility means
- Key Considerations for Evaluation and Implementation
- Patterns and Anti-Patterns Impacting Agility (Some Suggested Do's and Don'ts)



# About Me

- Director of Agile Consulting at LitheSpeed, Trainer and Agile Coach
- 25+ years in technology, throughout all areas of the SDLC
- 15+ years in senior leadership roles (S/VP, CIO, CTO)
- Oversaw/led successful implementation or material updating of nearly a dozen COTS systems in Financial, Insurance and other industries



(And yes, I do enjoy Scrabble.)

**John Halberstadt**

W

W

H

H

Y

Y

(are we here?)

- 1. To Debunk the Myth That Agile Approaches Are Incompatible With COTS Implementations*
- 2. To Provide Some Simple Techniques To Ensure Agility in Implementing COTS Systems*

# What Are COTS\* Systems?

*\*(Commercial, Off-the-Shelf)*



# Classification of COTS Systems

- **Closed** – generally configurable to address basic business needs, but not highly customizable and users must conform to existing features and functionality provided by the vendor
- **Open** – material changes can be made, through code as well as configuration, to support unique organizational business needs, as a key part of the platform's infrastructure and architecture

***Our focus for  
this discussion  
is on Open COTS  
systems.***



# Some COTS Platforms You May Already Use





# Poll and Discussion

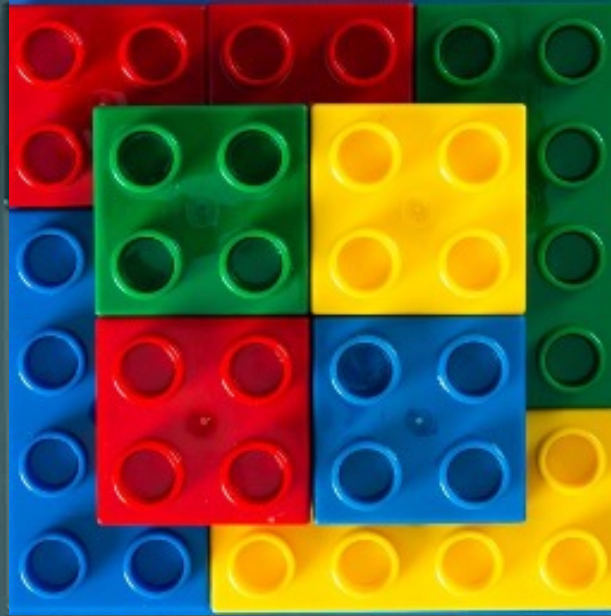
## Who:

- Has completed a COTS implementation?
- Is actively implementing a COTS system?
- Will be implementing a COTS system in the future?



# It's COTS – Why Would We Have To Customize?

- COTS platforms generally offer genericized or simplified features and products “out of the box”
- COTS systems’ “out of the box” usability, with no configuration, integration or customization is effectively impossible for many organizations, especially mid- to large-sized enterprises



**Your organization is likely to have specialized business-specific requirements, including**

- Unique workflows
- Specialized products or services
- Streamlined or more sophisticated user interfaces and user journeys, goals and experiences
- Integrations with third party providers or platforms – data, reporting, etc.

A man with glasses is shown in profile, looking towards a wall covered in colorful sticky notes. The sticky notes are in shades of blue, yellow, and green. The man is wearing a dark blue shirt with a red and white striped collar.

# What is Agile, What is Agility?

- To define how we can be agile in implementation, we need to align on what agility means, at the highest level
- Irrespective of technology, methodology or framework, we need to ensure we are:
  - Customer (and value) focused
  - Providing quantifiable business value
  - Being adaptable to changes that emerge
  - Working collaboratively, as a Team
  - Engaging our customers and eliciting their feedback
  - Incorporating feedback, continuously, to improve our products and services
  - Delivering early and often
  - Enabling predictability and consistency in throughput
  - Ensuring quality

# MIND THE GAP

## Some Key Differences in COTS Implementations



# Unique Considerations for COTS Systems



## Core Platform Sufficiency for Core Business Needs

- From a business functionality perspective, does the foundational platform support core products, services and capabilities



## Ability to Customize – Effort, Time and Expense

- There is a tipping point where excess customization of a COTS system negates the value of the system, while inflating expense and overhead



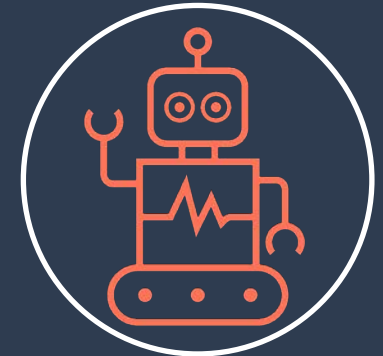
## Integrations and Third-Party Add-Ins

- Mature platforms' ecosystems offer modular extensibility; additional expense but relatively limited effort, time and risk



## Configuration versus Development

- Understanding what can be configured versus what must be developed through code is essential in the overall assessment process – for current as well as future needs

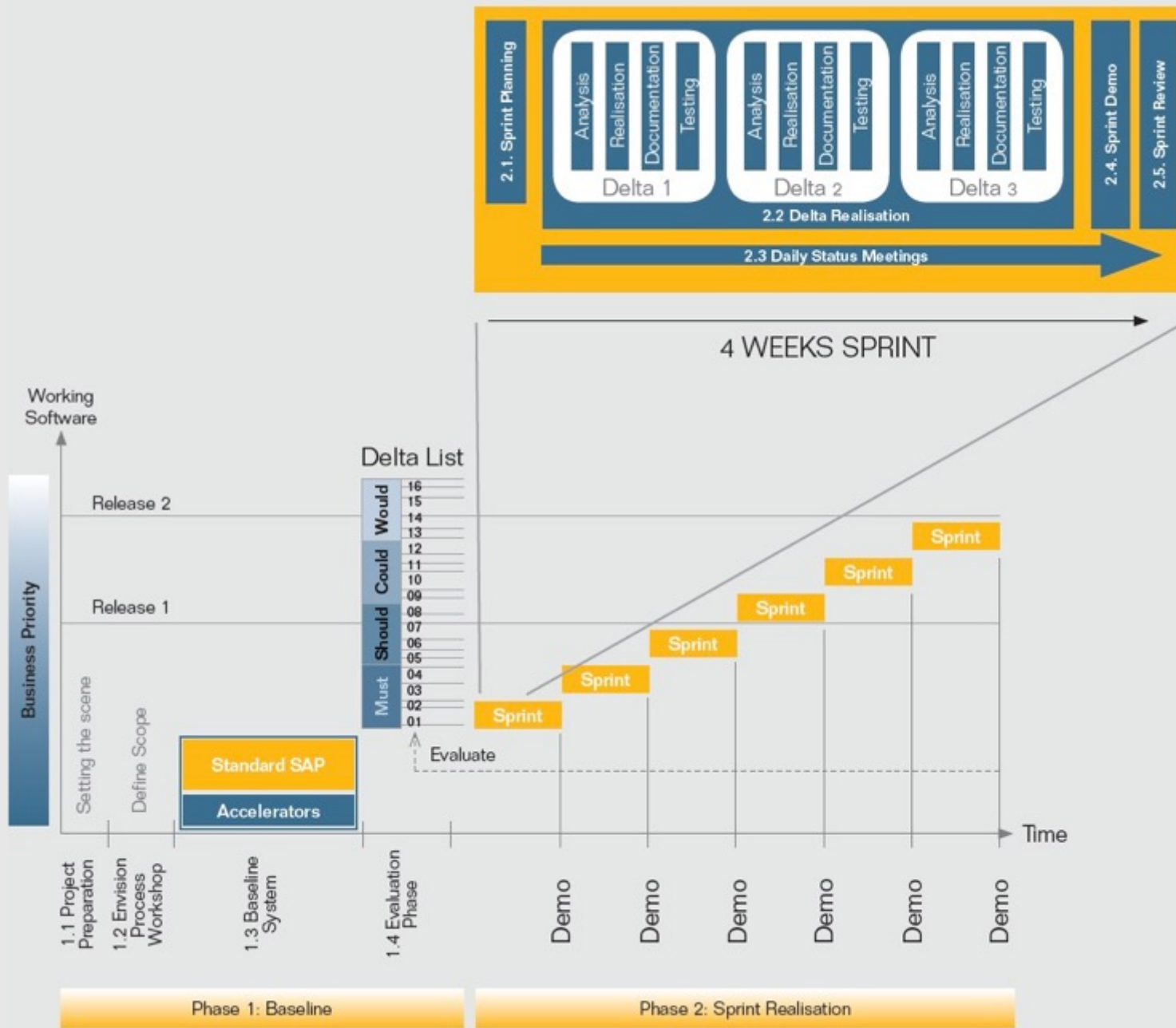


## Technical architecture and requirements

- For “on prem” solutions, as well as some SaaS offerings, core platform architecture requirements can be challenging if not part of the current enterprise ecosystem (operating system, database platforms, document repositories, etc.)



# The Big Picture, With Agility



<https://www.scruminc.com/implementing-sap-with-scrum/>

(c) 2022 TheSpeed

- SAP’s example timeline reflects some stages that we will review in more detail later in the discussion
- Note that the initial phase includes foundational steps for the “out of the box” system to be in place first
  - The core system
  - Accelerators, plug-ins, modules or additional “off-the-shelf” components
- This approach enables short iterations and early working software
- This foundation allows for interaction and user feedback and evolving requirements



# Evaluation Phase

*Considerations for COTS Systems*



# Overall Requirements Gathering Challenges

## Discussion:

- What makes requirements gathering more difficult with COTS than proprietary development?
- What assumptions do we have that we may need to re/examine regarding implementing these systems?
- What requires more discovery, analysis and documentation than with proprietary development initiatives?
- What efforts can be lessened with COTS implementations as compared to proprietary development initiatives?
- What techniques do we use today in our agile approaches that could be helpful?



# In the Evaluation Phase

*Choosing a platform can still be done with agility, however...*

The evaluation process with COTS systems does require some additional focus and planning, because:

- Unlike proprietary application development, pivoting can be challenging if the issue is inadequate or missing core functionality
- If the core system reasonably cannot support organizational needs for performance, scalability, security, interoperability, these are effectively impossible to “build in”

*A business-value/capability focus, documented in a prioritized MVP/MMP should be developed to help ensure a holistic understanding of what is essential for the platform to be, do, and support.*



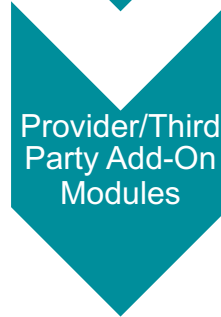
# In the Evaluation Phase

When assessing COTS systems, it is essential to understand:

- What functionality is core, “out of the box”
- What level of customization is available via configuration
- What integrations, add-ons or third-party tools exist within the vendor ecosystem
- What can be extended or modified via code, and in what programming language(s)



- Preferred approach whenever possible
- Best when able to be performed by business staff or proxies, e.g. business analysts for non-technical functions
- Should be well-documented, training available and mature from a platform perspective



- Many vendor ecosystems provide additional "add-ons" that extend existing functionality
- If supported and certified by the platform vendor, this generally is preferable to custom development in cost, risk and maintenance efforts
- As they are not part of the core package, there is generally an additional expense

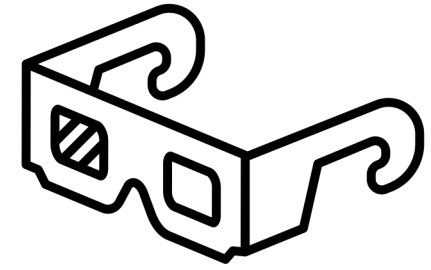


- Similar to provider or third-party add-ons or modules but generally at the data/API level
- Core platform User Interface and User Experience impacts should be assessed to limit negative impacts to the user through the integration
- These generally also will be desired or required for internal systems' data transfer, and some custom development for internal legacy systems to support may be required, e.g. to create an API rather than FTPing CSV files or direct database access



- Often required to support unique organizational needs BUT
- Should be the last option after the above have been exhausted
- Excess custom development is not only expensive in financial terms, but opportunity cost, time and upgrade effort

# Evaluation Phase: RFP/RFI



Understand foundational needs and strategic business goals and ensure the platform is able to support them

Verify reference customers' time to pilot and full production, as well as resource model – empirically-based data on the real effort

Ask reference customers specifics as far as how much was used “out of the box” versus highly customized

A low ratio of “out of the box” functionality may indicate a disconnect with core business processes, practices and needs

# Evaluation Phase: Final Considerations



- Remember that one often underappreciated value of off-the-shelf systems is that they generally have not only capabilities, but foundational starting components for use, such as:
  - Built-in templates
  - Products
  - Workflows
  - Reporting / Dashboards / Analytics
- Which have often been iteratively improved by many years of development based on many customers' needs in that domain
- Consider whether their “out of the box” functionality may be more effective for your organization than customizing to match your current processes, workflows and similar
- Understand the vendor ecosystem and extensibility to avoid being locked into a system that won't meet future business needs, if not current

# Implementation Phase

*Key Considerations for COTS Systems*



# Configuration

A hammer with a black head and a metal handle is on the left. A silver adjustable wrench is on the right. The background is a solid orange color.

Start here...

Configuration is the preferred choice when available to implement business user needs

Each system has unique requirements and approaches for configuration, some easier than others

Configuration to support requirements by business users or proxies, like the Product Owner or Business Analysts are preferable over requiring configuration by developers or system administrators

# Integration and Plugins / Modules

**Integrations likely  
require some  
development or  
sysadmin support  
to implement, and  
cost money, but...**



These integrations and plugins can extend system capabilities more quickly than traditional development

As an extension of the COTS model, they are generally validated to be functional with the platform, have a larger user base, and have been improved over time and feedback

They also often have their own roadmaps, and additional features are added more frequently than the core COTS system

# Custom Development

Custom development is often still required, even after configuration, integrations and modules

Approach custom development thoughtfully, and only use when necessary

Consider the impacts of core system upgrades, and whether any custom development may cause challenges in the future

This is especially the case if any core “out of the box” functionality is being impacted

# Potential Approaches with Agility





# Poll and Discussion: Ways of Introducing Agility

How would/have you introduced agility into an off-the-shelf system implementation?

- Engage business users early and often
- Evolve requirements throughout the process as they emerge
- Pilot a simplified, out-of-the-box implementation to get user feedback quickly
- Deliver components of the system for business use incrementally by user area, type or similar
- Other – please share!





# Slices Versus Layers

As with traditional agile development, ensure that each product increment is focused on business value, irrespective of the “how”

Some value may come from simply enabling basic features “out of the box” for early business use and usability feedback


Some stories and functionality may simply be configuring “out of the box” features

”How” the functionality is enabled for the users is less important than that it is a business valuable piece of functionality being made available

Start small and simple

Get feedback and iterate changes, whether through configuration or code

# Slicing Methods to Consider

- 
- By module/component
  - By feature (OOB, limited or no customization)
  - By state, region, country
  - By user type or role (standard, admin)
  - Add custom development (behaviors, UI/UX improvements) iteratively, and only as needed - remember that COTS platforms' value is in what is “out of the box”
  - Adding integrations once core functionality is in place (note: some integrations are required for core functionality, so analysis needs to be performed)

# Suggested Do's

- Do assess needs from a functional rather than technical foundation first to ensure core features and functionality meet anticipated needs
- Do ask targeted questions to reference customers related to time/effort/expense for initial implementation and upgrades
- Do understand the vendor roadmap and technical extensibility
  - What is the vendor ecosystem – modules, add-ons, etc.
  - What is available via integration – data in/out
  - What is configurable – workflow, products, etc.
  - Is configuration done by developers or can it be done by analysts or business-focused staff
  - What can/must be done via code/traditional development
  - What language(s) are supported – proprietary or commercially available
  - What core infrastructure options exist/are required (database, reporting, document storage, etc.)
- Do iterate through implementation to get a “thin slice”, business usable increment of functionality stood up as quickly as possible



# Suggested Don'ts

- Don't define exhaustive and prescriptive requirements for the full implementation initially
- Don't override foundational behavior, features and functionality solely to match current processes
- Don't do a “big bang”, monolithic implementation
- Don't customize via code until verified necessary that
  - Functionality is truly needed, not just legacy system capabilities
  - Configuration changes will not address the need
  - Additional vendor or third-party modules/add-ons will not address the need
- Don't forget organizational change management
  - Leverage vendor materials and resources where possible
  - As with the system itself, customize only when there are compelling business reasons to do so



**Q & A & Thank You!**



# CONTACT

John Halberstadt  
[john.halberstadt@LitheSpeed.com](mailto:john.halberstadt@LitheSpeed.com)

[lithespeed.com](http://lithespeed.com)

+1 (703) 745-9125  
@LitheSpeed



[LitheSpeed.com](http://LitheSpeed.com)