Data Collection for Agile Projects

Blaze Smallwood

ICEAA Conference 2016

Table Of Contents

- Background: Agile Software Development Methodology
- Types of Data and Data Sources
- How Data is Used

What is “Agile” software development?

- What is “Agile” Software Development?
  - A software development philosophy based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams
  - Promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change
- Agile Principles
  - Customer satisfaction through early and continuous delivery of valuable software
  - Welcoming changing requirements, even late in development
  - Deliver working software frequently
  - Working software is the primary measure of progress

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

**Manifesto for Agile Software Development © 2001*

Agile Concepts and Terms

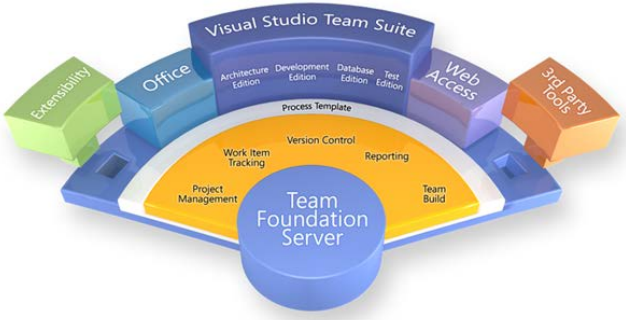
- **User Stories**: A high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement
- **Complexity Points**: Quantification of a User story's scope or effort, a relative measure of complexity
- **Sprint / Iteration / Release**: *Sprint* - Fixed time-box in which development occurs (usually 2 - 4 weeks); *Iteration* - Minor subset of requirements designed to be released to the user community; *Release* - Multiple Iterations that fulfill a major subset of user requirements
- **Velocity:** Performance / productivity measure that indicates progress toward capability delivery (i.e., Complexity Points completed per sprint)
- **Project / Sprint Backlog:** A prioritized database that summarizes the User Stories / Requirements yet to be complete for the entire project
- **Burndown:** The concept, often shown as a graph over time, of working off or "earning" Complexity Points toward iteration or delivery completion

Table Of Contents

- Background: Agile Software Development Methodology
- Types of Data and Data Sources
- How Data is Used

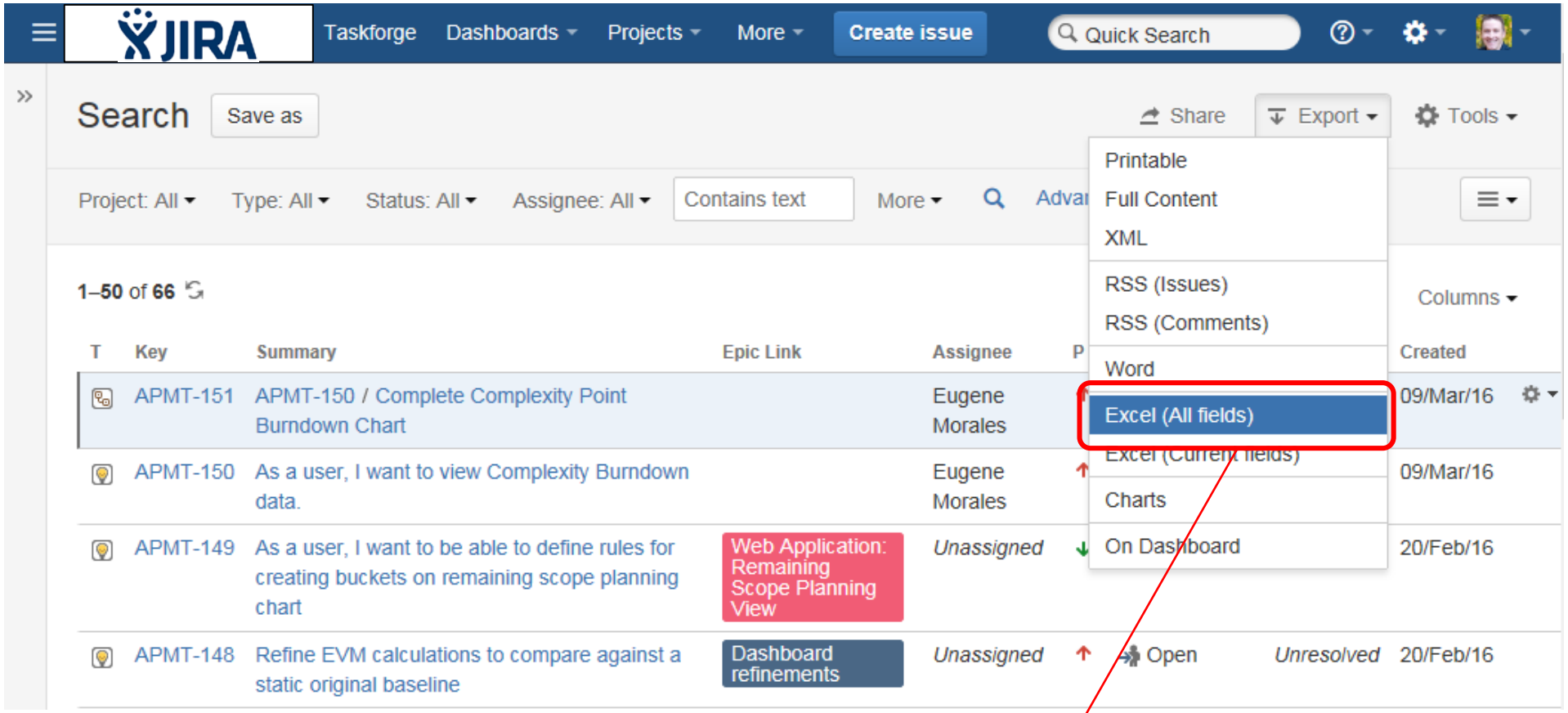
Sources to Collect Agile Metrics Data - Overview

- Access to contractor's Application Lifecycle Management (ALM) tool
  - Best method to gain insight on a developer's backlogs
  - Popular ALMs include: JIRA, Forge, or Microsoft Team Foundation Server (TFS)
- CDRL on contract with software developer to report on a monthly basis
  - Supporting documentation provides a template to capture information that can be found in an ALM
  - Similar to SLOC delivered in SRDR



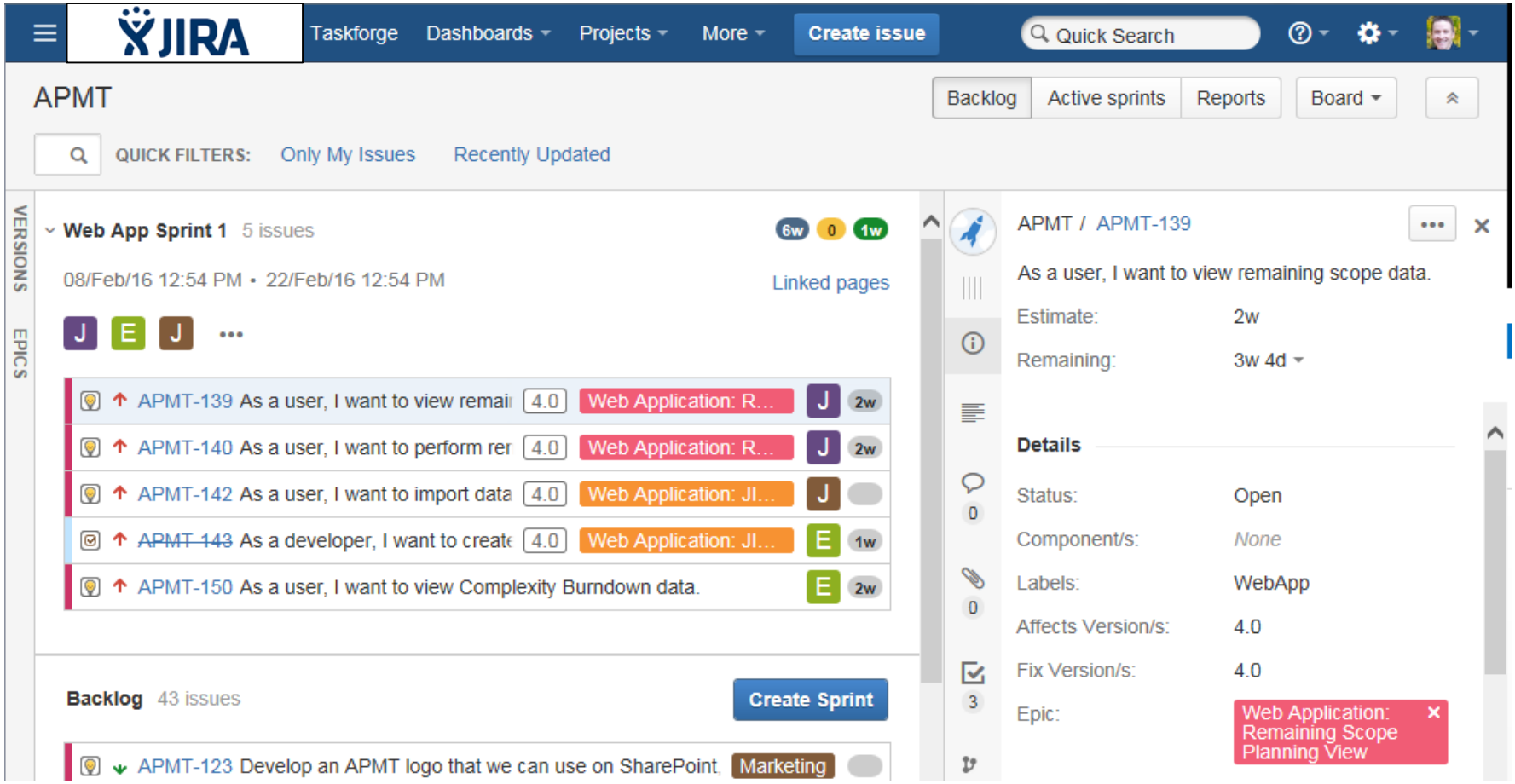
© Can Stock Photo

Sources to Collect Agile Metrics Data – ALM (Data Export)



Can export issues list into Excel format with all fields from JIRA

Sources to Collect Agile Metrics Data – ALM (Boards)



- Provides detailed insight into scope currently being worked and on the backlog
- Can search for individual issues/artifacts and see details on each

Sources to Collect Agile Metrics Data – CDRL Template

- Block 16 of the CDRL template should capture all the directions to instruct contractor on specific information to provide

CONTRACT DATA REQUIREMENTS LIST
(1 Data Item)

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 110 hours per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503. Please DO NOT RETURN your form to either of these addresses. Send completed form to the Government Issuing Contracting Officer for the Contract/PR No. listed in Block E.

A. CONTRACT LINE ITEM NO.	B. EXHIBIT	C. CATEGORY:
	A	TDP ____ TM ____ OTHER __X_____

D. SYSTEM/ITEM	E. CONTRACT/PR NO.	F. CONTRACTOR
Program XYZ	N000XX-15-C-0222	Contractor ABC

1. DATA ITEM NO	2. TITLE OF DATA ITEM	3. SUBTITLE
A018	Agile Software Metrics Report	

4. AUTHORITY (Data Acquisition Document No.)	5. CONTRACT REFERENCE	6. REQUIRING OFFICE
DI-MGMT-81740A and See Block 16	SOW 3.1.1.1.1, 3.2.3.3	PMW / XYZ Division

7. DD250 REQ	9. DIST. STATEMENT REQUIRED	10. FREQUENCY	12. DATE OF FIRST SUBMISSION
LT	C	See BLK 16	See BLK 16
8. APP CODE		**11. AS OF DATE**	**13. DATE OF SUBSEQUENT SUBMISSION**
A		N/A	See BLK 16

16. REMARKS:

The contractor shall answer each of the following questions using the "Agile Software Metrics Report Template" Microsoft Excel file:

General (Information provided on the INPUTS tab):
Table 1: Planned Number of Weeks per Sprint
Table 2: Point Scale Definition
Define and describe the point scale values used to describe the complexity of the effort to complete a given artifact.
Table 3: List of Artifact Types
List and describe the various artifacts that are planned on being executed for the entire project.

14. DISTRIBUTION			
a. ADDRESSEE	b. COPIES		
	DRAFT	FINAL	
		Reg	Repro
COR	1	1	
ENGR	1	1	

Sources to Collect Agile Metrics Data – CDRL Supporting Template

Development History

Agile Software Metrics Report

Description and Instructions for providing input to the "Artifacts by Sprint" table

The "Artifacts by Sprint" table provides an historical catalogue of all artifacts addressed in each Sprint

1. "Artifact ID": Captures the unique identification code for the artifact
2. "Artifact Description": Provides a brief description of the artifact
3. "Artifact Type" references Table 3 on 'Inputs' list of artifact types. Examples of artifacts include Story, IA, HSI, Test, Defect, Peer Review, etc…
4. "Sprint Started": Specifies the Sprint in which work on the artifact was started
5. "Sprint Completed": Specifies the Sprint that the artifact was completed. If an artifact has not been completed by the reporting date of the CDRL, leave blank or indicate with "N/A".
6. "Pop Up Artifact?": Select "Y" if the artifact was an unplanned artifact and "N" if the artifact was planned for prior to the start of the Sprint
7. "Requirement ID": Captures the unique identification code for the requirement being addressed by the stated artifact
8. "Does this Artifact Complete the Requirement?": Select "Y" if the completion of the stated artifact completes the requirement referenced in the "Requirement ID" column and "N" if the completion of the stated artifact does not complete the requirement
9. "Actual Complexity Points": Provide the actual complexity points associated with the stated artifact

Artifacts by Sprint

#	Artifact ID	Artifact Description	Artifact Type	Sprint Started	Sprint Completed	Pop Up Artifact?	Requirement ID	Does this Artifact Complete the Requirement?	Actual Complexity Points
1									
2									

- Development history captures all issues completed to date. Other information includes:
  - Start/end dates of when issues were worked on (end date left blank if still in progress)
  - Estimated story point for each issue
  - How issues map to requirements
- This table is updated periodically as work is completed each sprint

Sources to Collect Data – CDRL Supporting Template

Project Backlog

Description and Instructions for providing input to the "Project Backlog Data" table

The "Project Backlog Data" table provides a current capture of the full project backlog, which includes relevant data for all artifacts currently planned to be completed in the future

1. "Artifact ID": Captures the unique identification code for the artifact
2. "Artifact Description": Provides brief description of the artifact
3. "Artifact Type": Provides a category for the artifact. Examples of artifact types include Story, IA, HSI, Test, Defect, Peer Review, etc…
4. "Priority": Specifies the current priority rating of the artifact based on a separately defined scale agreed upon by the Program Office (e.g. "1 - Urgent", "2 - High", "3 - Moderate", "4 - Low")
5. "Status": Specifies current status of the artifact (e.g. "Closed", "In Progress", "Not Started", etc.)
6. "Points or Hours Estimate": Captures current complexity point or hours estimate for the artifact

Project Backlog Data

#	Artifact ID	Artifact Description	Artifact Type	Priority	Status	Points or Hours Estimate
1						
2						

- The project backlog captures the work remaining to complete the project. Other information provided in the project backlog include:
  - Issue description, artifact type, priority, status, story point estimate
- Issues from this table will be move to the development history table as they are completed
- Captures entire scope of the project up front, as backlog changes over time, issues get added and scope could grow in addition to the original plan

Sources to Collect Financial Data

- Enterprise Resource Planning (ERP) systems
- CDRLs on a Contract
  - Table within a custom Agile Metrics CDRL
  - Other program management CDRLs (Monthly Status Report, Integrated Program Management Report)

CONTRACT DATA REQUIREMENTS LIST (1 Data Item)															Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 110 hours per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503. Please DO NOT RETURN your form to either of these addresses. Send completed form to the Government Issuing Contracting Officer for the Contract/PR No. listed in Block E.														
A. CONTRACT LINE ITEM NO. TBD			B. EXHIBIT A			C. CATEGORY: TDP ____ TM ____ OTHER __X_____								
D. SYSTEM/ITEM Program XYZ			E. CONTRACT/PR NO. TBD			F. CONTRACTOR TBD								
1. DATA ITEM NO A001			2. TITLE OF DATA ITEM Monthly Status Report			3. SUBTITLE								
4. AUTHORITY (Data Acquisition Document No.) DI-MGMT-80368A and See Block 16			5. CONTRACT REFERENCE TBD			6. REQUIRING OFFICE PMO XYZ / Division ABC								
7. DD250 REQ LT		9. DIST. STATEMENT REQUIRED C		10. FREQUENCY See BLK 16		12. DATE OF FIRST SUBMISSION See BLK 16			14. DISTRIBUTION					
8. APP CODE A				11. AS OF DATE N/A		13. DATE OF SUBSEQUENT SUBMISSION See BLK 16			a. ADDRESSEE	b. COPIES				
										DRAFT	FINAL			
											Reg	Repro		

Source to Collect Financial Data – Custom Agile CDRL

Description and Instructions for providing input to the "Sprint History" table

The "Sprint History" table provides a financial and resource summary for each Sprint

1. "Sprint #": Specifies the Sprint number for each Sprint completed to date
2. "Date Started": Provide the date the Sprint started
3. "Date Completed": Provide the date the Sprint completed
4. "Team Size": Provide the Scrum team size in terms of full-time equivalents
5. "Hours Expended": Provide the actual hours expended by the Scrum team during the Sprint
6. "Dollars Expended": Provide the actual labor cost expended during the Sprint

Sprint History

#	Sprint #	Date Started (MM/DD/YYYY)	Date Completed (MM/DD/YYYY)	Team Size (FTEs)	Hours Expended	Dollars Expended ($)
1						
2						

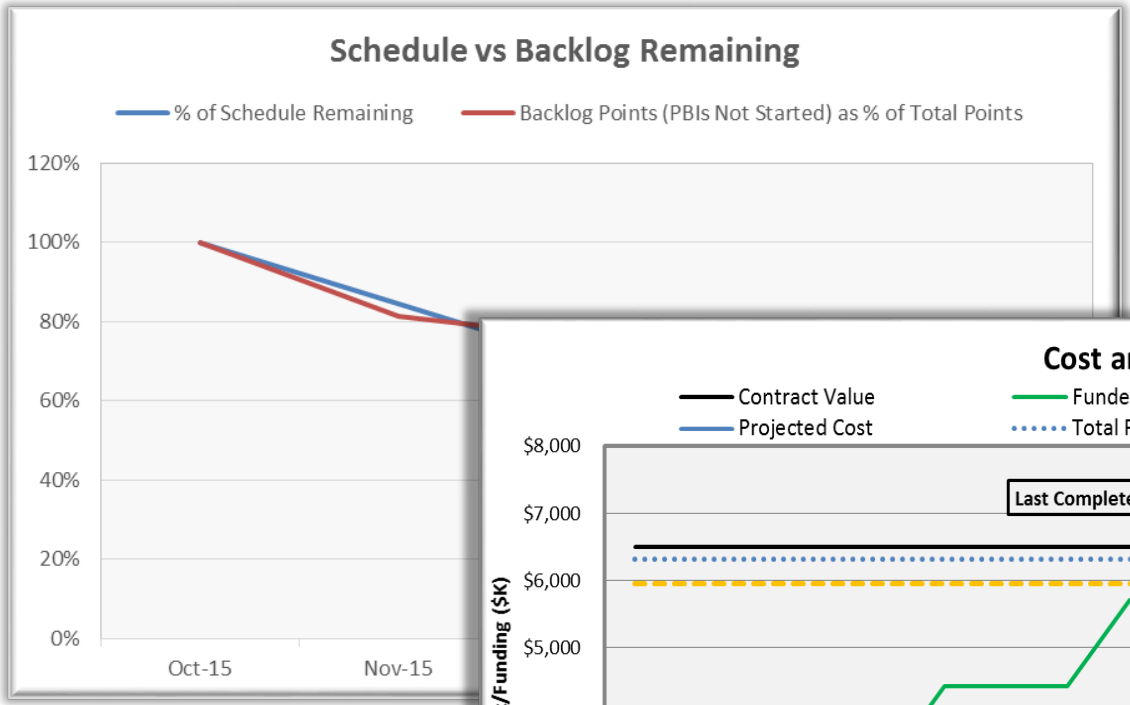
- The Sprint History table captures actuals costs expended to date on sprint by sprint basis. Other important information to capture include:
  - Start and end dates of sprints
  - Captures team size and hours expended
- This section should include total costs of the entire project, and not just the costs attributed to the software development efforts

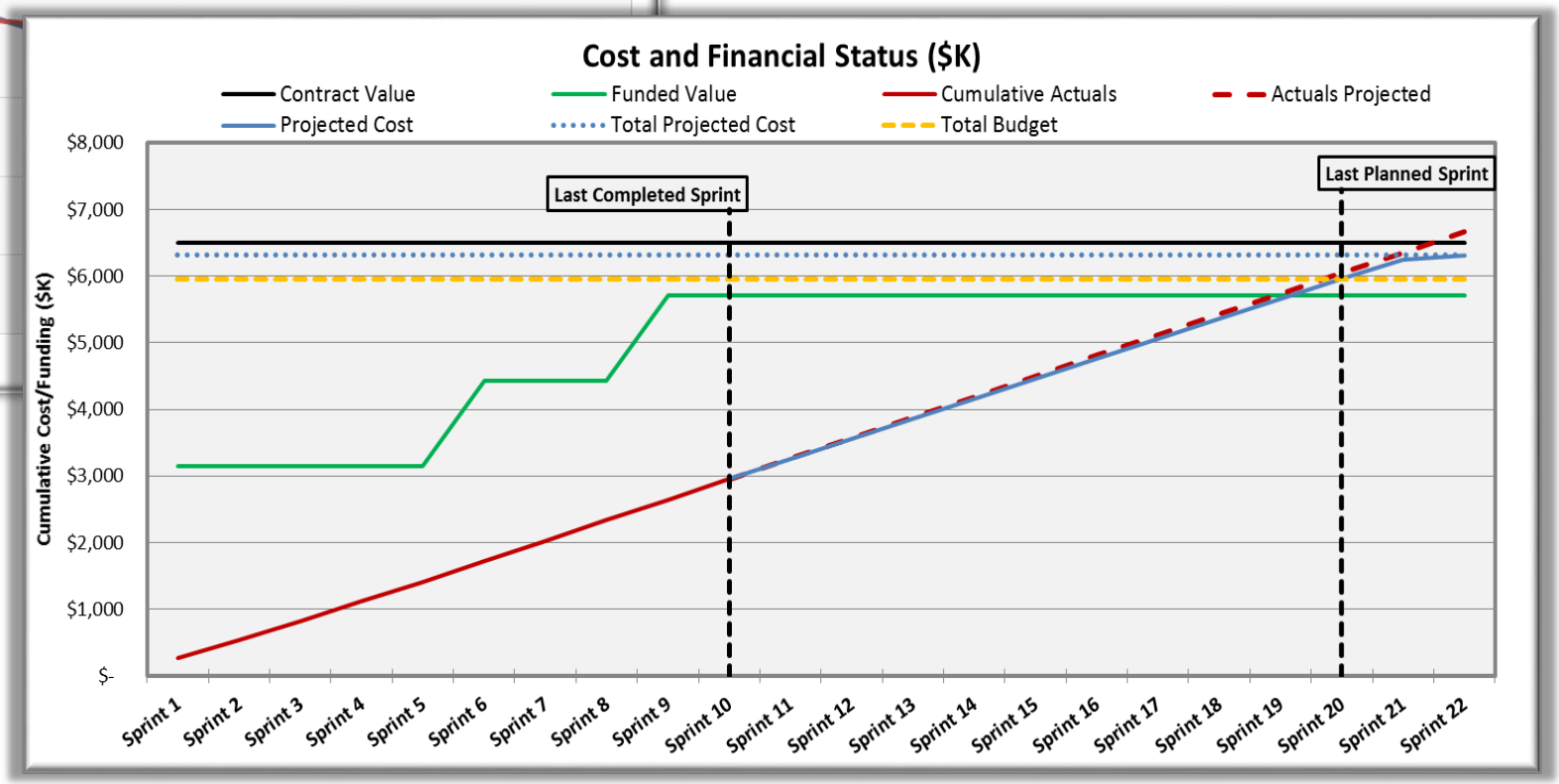
Table Of Contents

- Background: Agile Software Development Methodology
- Types of Data and Data Sources
- How Data is Used

Cost and Schedule Projections

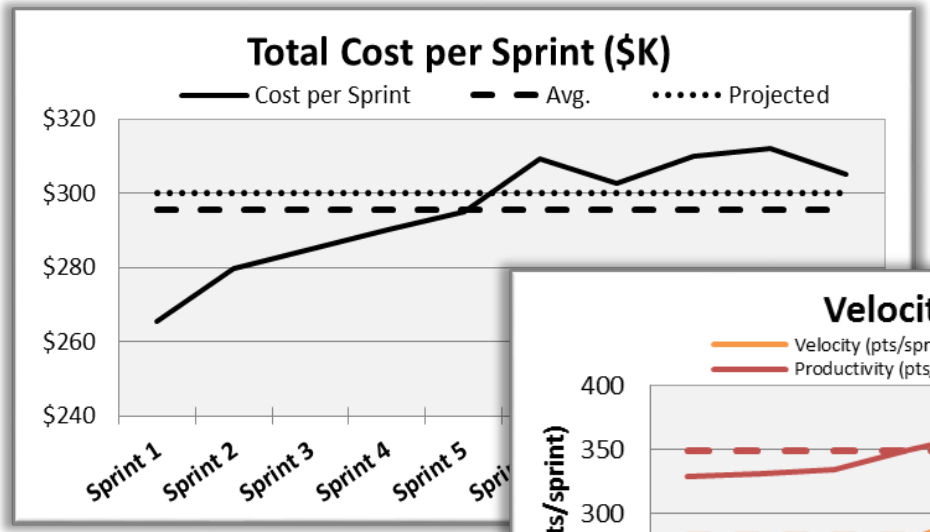
Data Source:
Financial data from CDRL, project backlog, and development history from CDRL or ALM

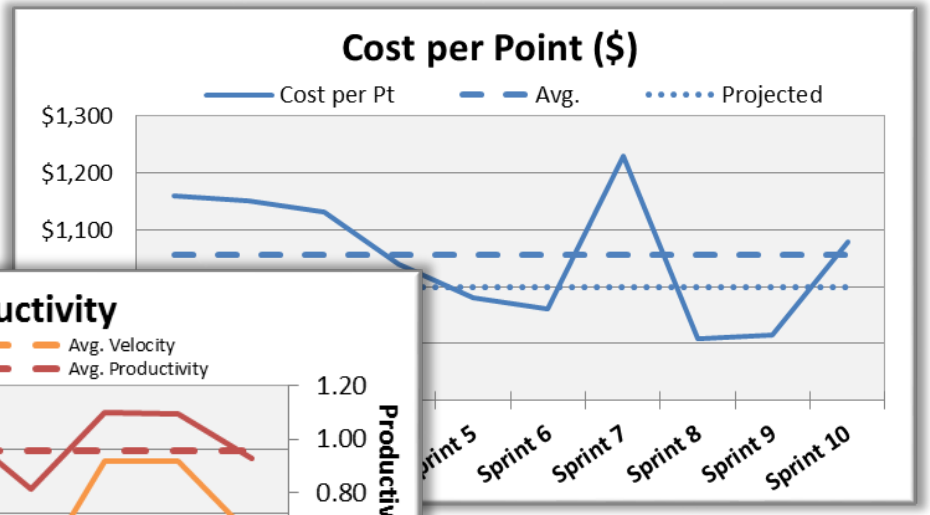


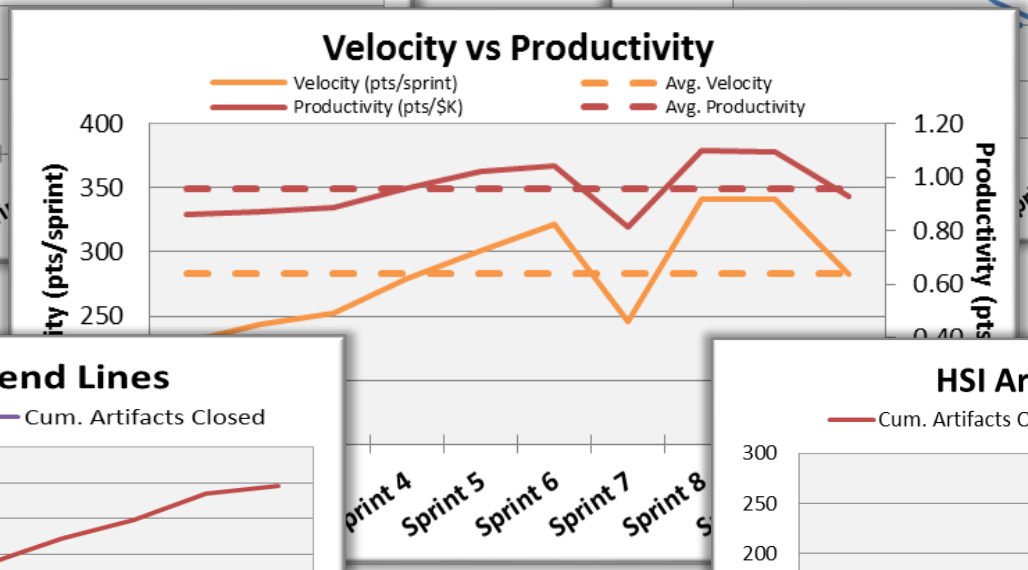


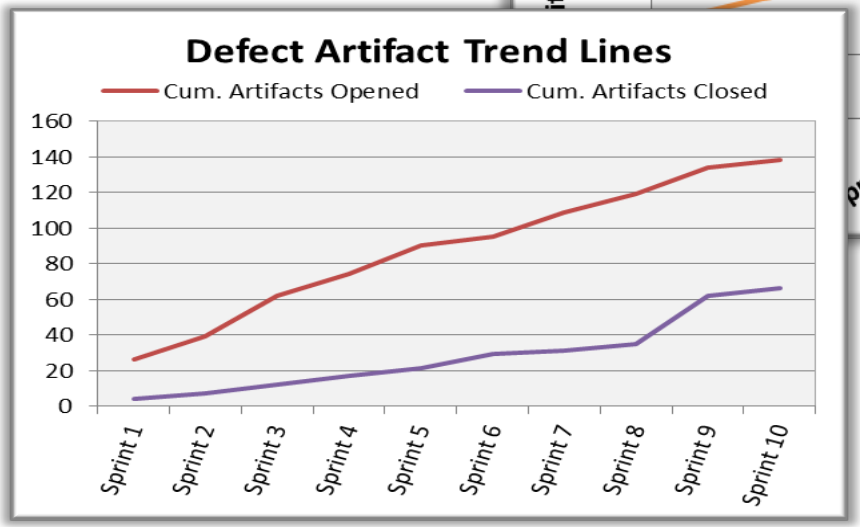
Historical Performance Analysis

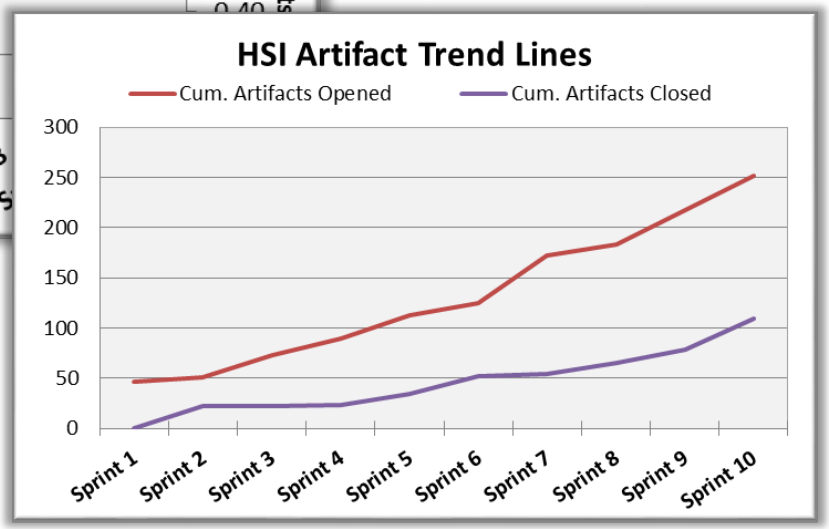
Data Source
Development history from CDRL or ALM



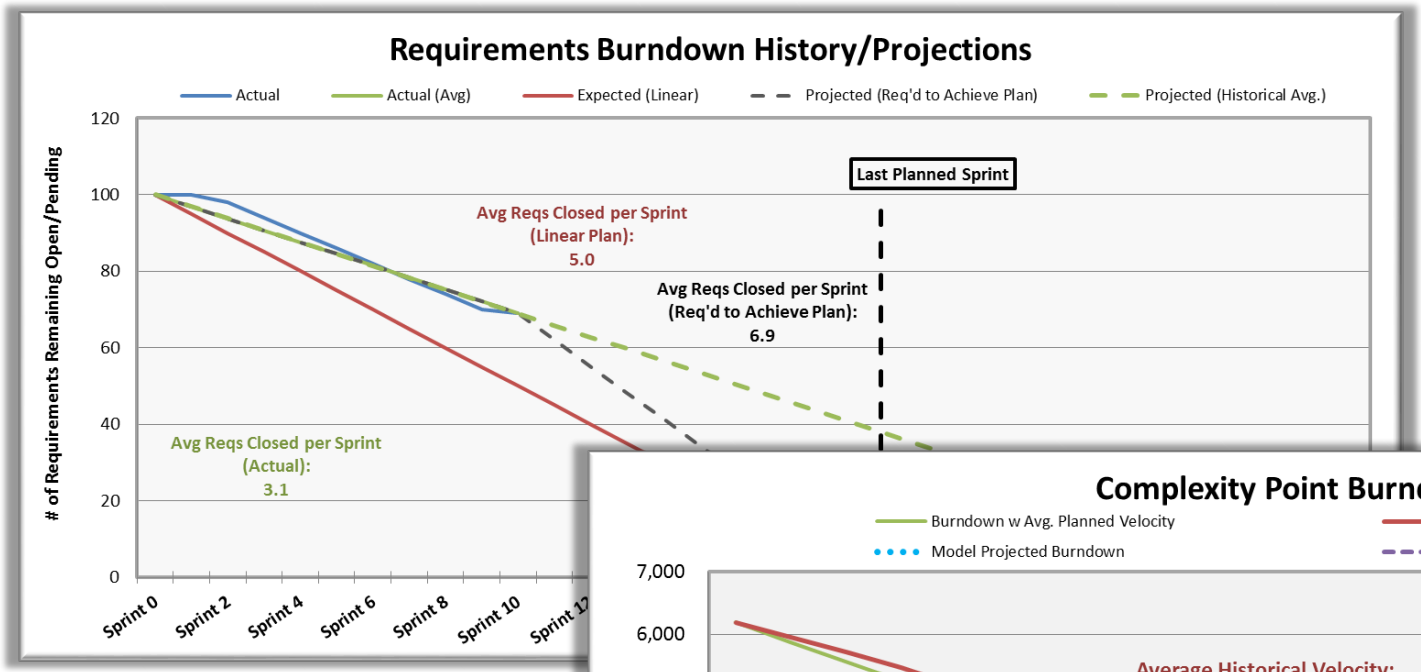


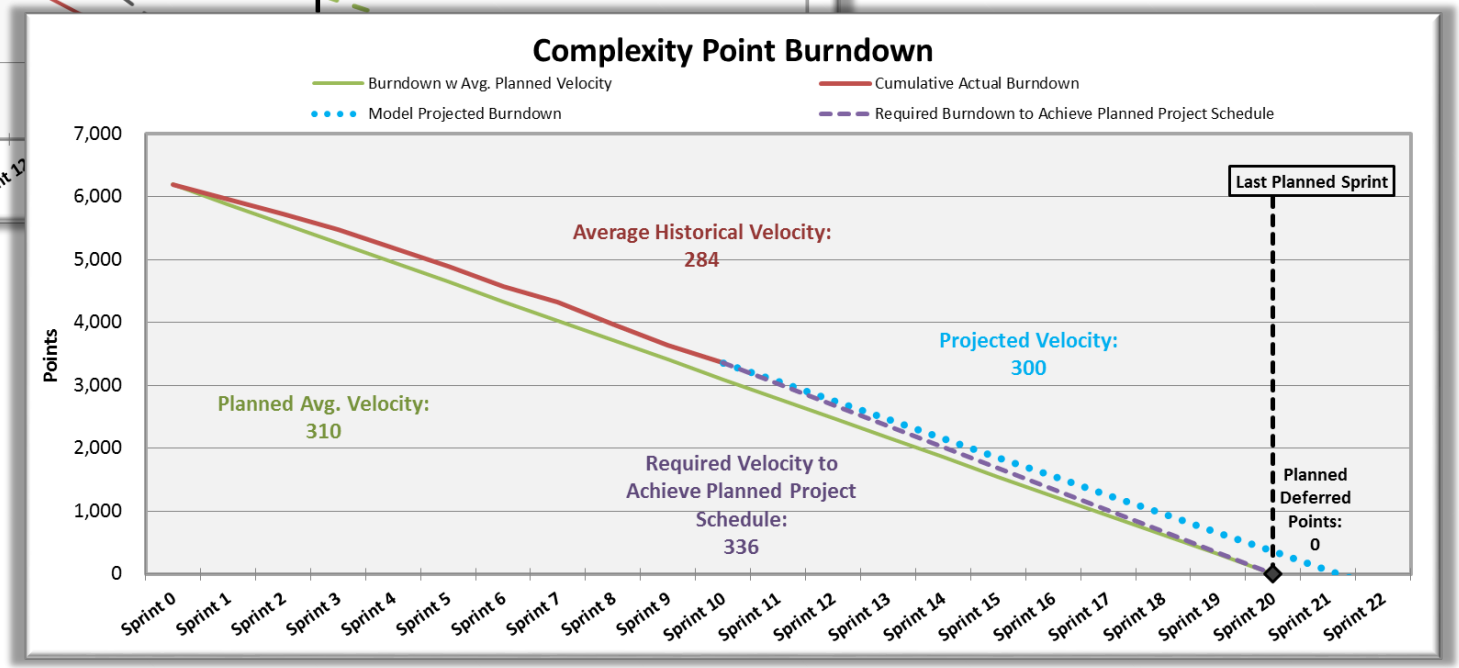






Requirements and Complexity Point Burndown and Projections





Data Source:
Development history from CDRL or ALM

Lessons Learned

- Best method to collect backlog data is access to a developer's ALM
- Be clear and concise with instructions in the CDRL and supporting template
- Require developer to provide an initial report that captures a baseline backlog for the full scope of the project
- Communicate as much as possible with developer to clarify ambiguity with the data
- Eliminate duplicative data collection
  - Leverage existing CDRLs to collect financial data
  - Do not require custom CDRL if developer provides access to ALM

Conclusion

- Prior CDRL and data collection methods have struggled to give insight into the progress of an Agile software development project
- A new set of data collection methods have been implemented on several projects and have helped give PMs of Agile projects increased insight

For further information . . .

Blaze Smallwood
Associate

Booz | Allen | Hamilton

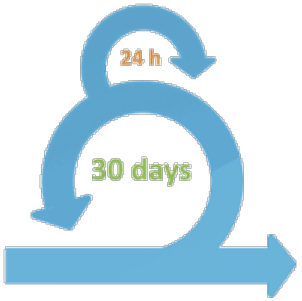
Booz Allen Hamilton Inc.
Office 309.359.3160
Mobile 619.850.6123
smallwood_blaze@bah.com

Back-up Slides

Industry trend towards agile software development methods presents challenges for PMs in managing cost, schedule, and scope

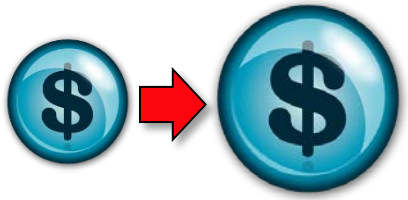
Agile Development

- Requirements flexibility
- Frequent delivery of software and interactions with product owners
- Scrum team defines processes to prioritize, estimate, and execute



PM Challenges

- Managing scope creep while maintaining flexibility
- Evaluating and reporting progress
- Projecting cost and schedule
- Prioritizing remaining work



Traditional vs. Agile Process Overview

Traditional (Waterfall) Approach

Sequential activity of one team

- Plan all of the requirements
- Design all of the requirements
- Develop all of the requirements
- Test all of the requirements

Users will receive end product once ALL requirements have been fully designed, developed, and tested

Agile Approach

Iterative approach where constant user interaction is preferred and highest priority items are completed first

- Determine arch/funct rqts
- Take each Iteration:
  - Design it, Develop it, Test it, Deploy it
- Each requirement can be designed, developed, and tested simultaneously along with other requirements

Agile doesn't change the end product, only the way projects are scoped, managed and executed