

Learning Analytics For The DL Courseware Factory

Analysis, Solutions, Approved Capability Requirements to Support Them,
and Way-ahead

Dr. Mitchell Bonnett

Note: The views presented are those of the speaker and do not necessarily represent the views of DoD or its components

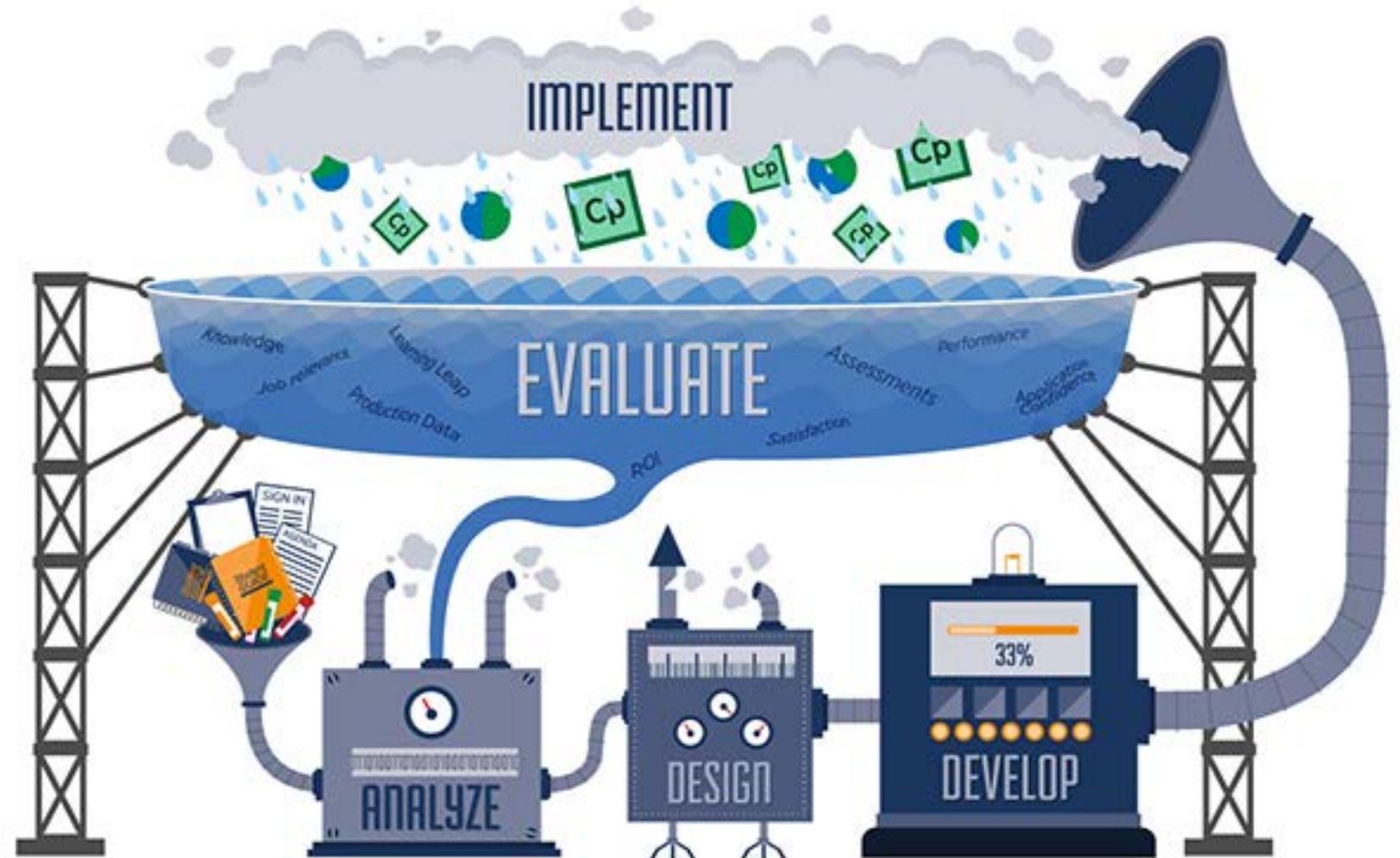
Introduction

The Analyze, Design, Develop, Implement, and Evaluate (ADDIE) model is the learning creation business process model for most of the industry.

When used to create course software (AKA courseware - CW) that is Computer Managed Instruction (CMI) at very large scale it's a business.

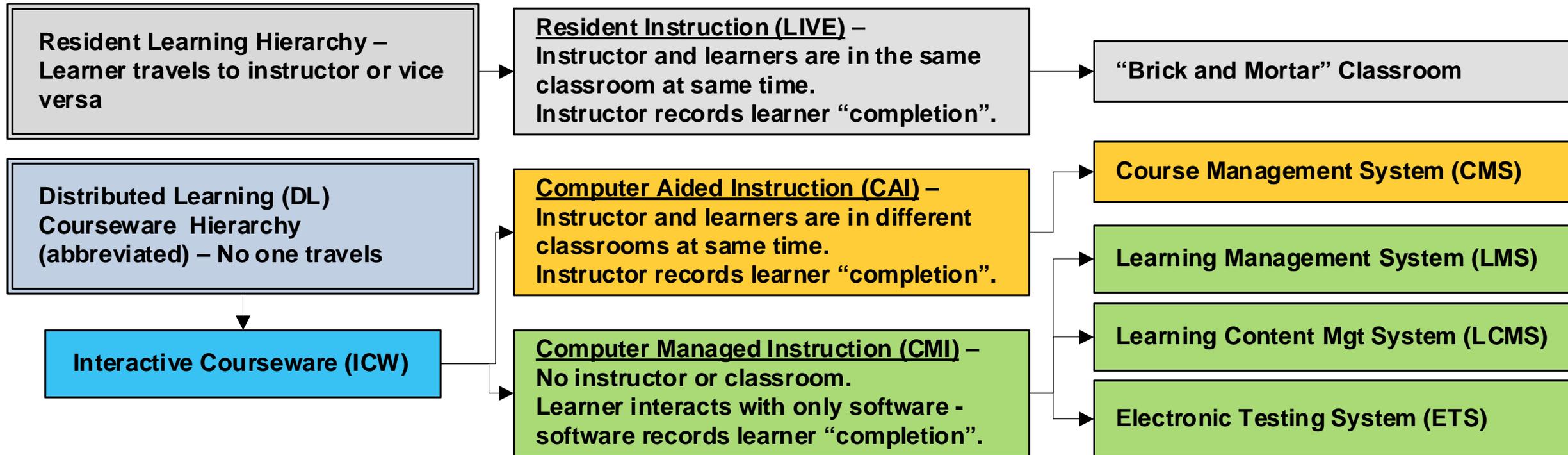
A courseware business.

A CMI factory.



Today's topic is factory measurement. Labor and tooling are future topics. Please hold questions to end. Thanks!

Scope: Computer Managed Instruction (CMI) IMI Type



- Focus today is CMI. Interactive courseware that is CMI is the most taken and completed DL IMI type. The discussion today is about rapid, efficient defect-free CMI production at scale.
- CMI is too often difficult to develop, implement, and evaluate – and it shouldn't be.

Scale: LARGE Scale CMI Courseware Development and Use

DEPARTMENT OF THE ARMY
Fiscal Year (FY) 2019 Budget Estimates
Operation and Maintenance, Army
Budget Activity 03: Training and Recruiting
Activity Group 32: Basic Skill and Advanced Training
Detail by Subactivity Group 324: Training Support



The Official Learning Development Site for the U.S. Federal Government

Distributed Learning (DL) Course Completed¹



- 1K+ CMI courses available in FY17
- 1M+ active learners took CMI courses in FY17
- 12M+ CMI completions in FY17

- ??? CMI courses available in FY17
- ??? active learners took courses in FY17
- 15M CMI completions in FY17

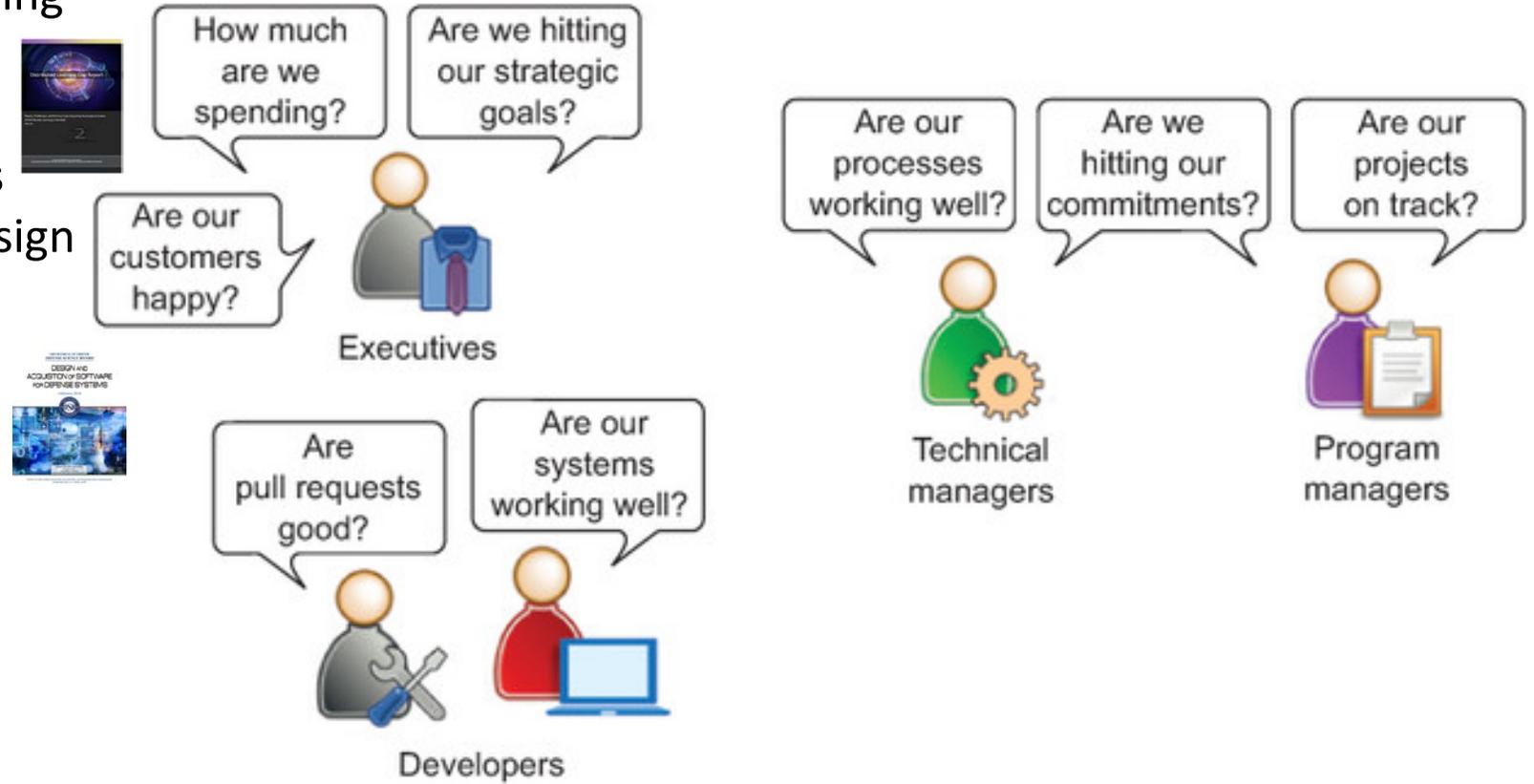
The Army uses SCORM 2004 3rd Edition to autoscore its CMI.

OPM uses SCORM 1.2 to autoscore its CMI.

Analysis - Recent History of the DoD Problem Set

External analysis finds DoD CMI factories need improvement and makes recommendations

- 2017 Advanced Distributed Learning DL Gap Report
 - Use Standards/Specs
 - Incorporate Learning Metrics
- 2018 Defense Science Board - Design and Acquisition of Software For Defense Systems
 - Transition to Factory
 - Go Agile
 - Use Agile Metrics
- 2018 OSD Reform Initiative - Learning Technology (LTech) Implementation Plan
 - Go Factory (USA Learning)



Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Yet... Will these achieve software industry level efficiencies in our business? Perhaps...

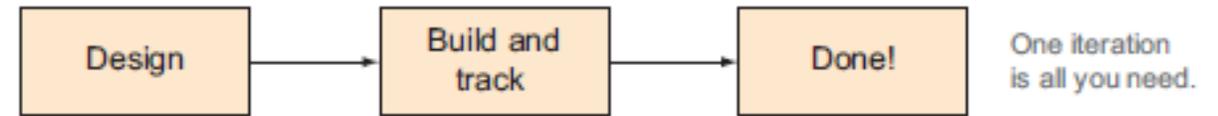
Analysis - Review of the Industry Problem Set

Analysis finds software industry may be unprepared for use of analytics (metrics) in Agile

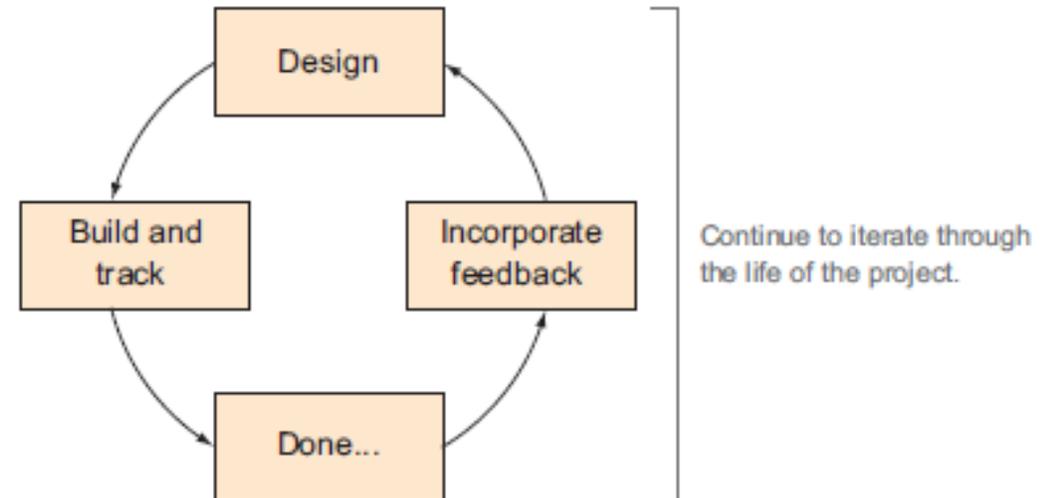
- 2015 Journal of Information and Software Technology “Using Metrics in Agile and Lean Software Development – A Systematic Literature Review of Industrial Studies” found:

- Agile focus on lightweight working practices, constant deliveries, and customer collaboration conflicts with Traditional measurement (metrics) approaches
- The overall picture is not clear on what metrics Agile teams are using in practice, for what purpose, and with what effect
- Projects and sprints need to be planned and tracked; Quality needs to be measured; and Process problems need to be identified and fixed

Project mentality



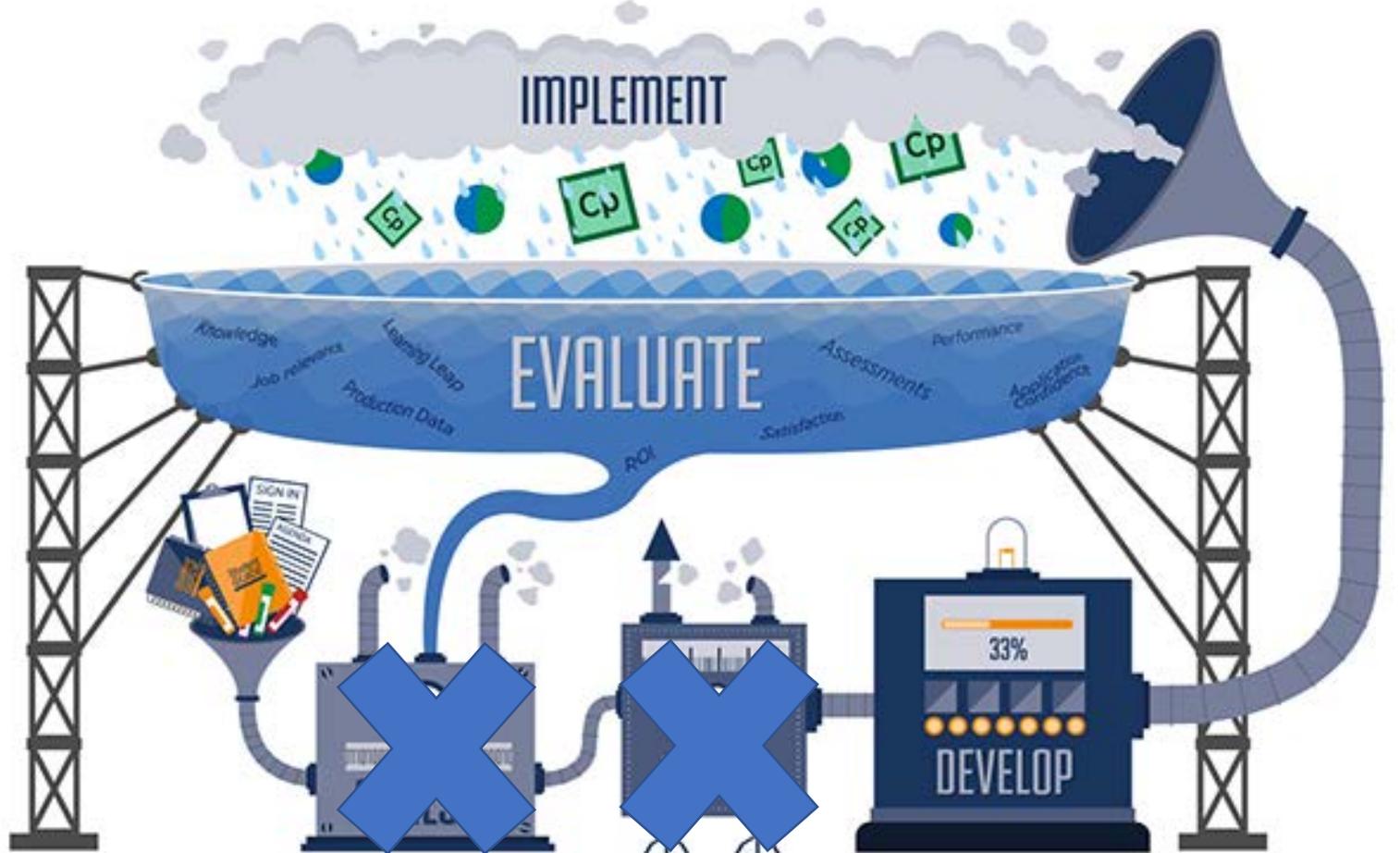
Product mentality



Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Analysis: ADDIE Stages that Affect the CMI Factory

- **Analysis** and **Design** stages do not affect CMI (its not DL yet)
- **Develop** stage is very strongly affected yet traditionally has the weakest analytics (metrics) reporting due to Develop stage span and hesitancy to make vendors brief internal practices
- **Implement** stage is strongly affected but trials and function tests can help force metrics up
- **Evaluation** stage is strongly affected but Learner Help Desk tickets can force metrics up



Analysis: High Influence Agile Industry Metrics

High influence metrics based on number of occurrences and perceived importance factor.

Metric	Number of occurrences	Importance factor	Sum of ranks/2
Velocity [S1,S2,S3,S5,S6,S8,S8,S10,S13,S16,S23,S27,S28]	15	3	1
Effort estimate [S3,S7,S8,S8,S9,S12,S15,S17,S29]	12	3	1.5
Customer satisfaction [S1,S3,S7,S17,S19,S20]	6	3	2.5
Defect count [S1,S3,S5,S7,S7,S10,S25,S27]	8	2	5
Technical debt [S4,S4]	2	3	5
Build status [S4,S14]	2	3	5
Progress as working code [S30]	1	3	6.5
Lead time [S18,S19,S22,S24]	4	2	7
Story flow percentage [S13]	1	2	9.5
Velocity of elaborating features [S13]	1	2	9.5
Story percent complete [S29]	1	2	9.5
Number of test cases [S1]	1	2	9.5
Queue time [S18]	1	2	9.5
Processing time [S18]	1	2	9.5
Defect trend indicator [S25]	1	2	9.5
Work in progress [S17,S20,S21,S22,S23,S24]	6	1	10
Number of unit tests [S1,S5,S14,S27,S28]	5	1	11
Cost types [S21]	1	1	14
Variance in handovers [S21]	1	1	14
Deferred defects [S7]	1	1	14
Predicted number of defects in backlog [S25]	1	1	14
Test coverage [S14]	1	1	14
Test-growth ratio [S14]	1	1	14
Check-ins per day [S5,S27,S28]	3	NA	16
Cycle time [S17,S23]	2	NA	16.5

No need to strain your eyes – they will be discussed in upcoming slides.

Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies. *Information and Software Technology*, 62, 143-163. doi: <https://doi.org/10.1016/j.infsof.2015.02.005>

Analysis: Where We Need to Be in Agile Metrics

Where we need to improve for Agile (all of us?)

Where we are (some of us?)

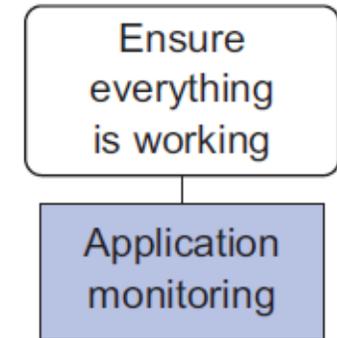
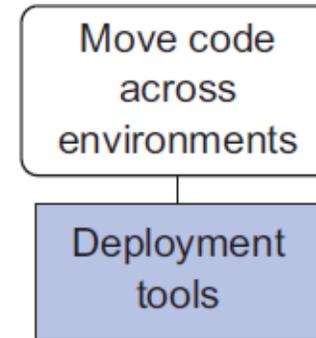
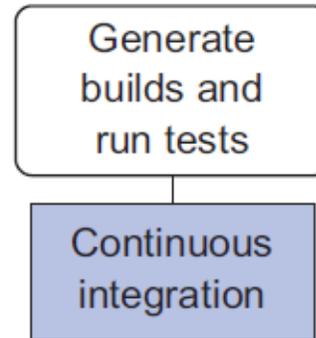
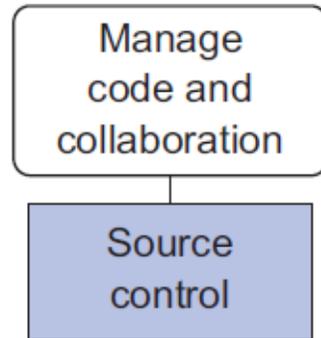
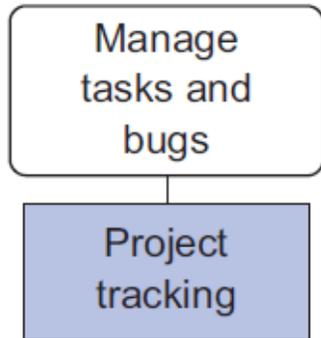
ADDIE
Develop

ADDIE
Develop

ADDIE
Develop

ADDIE
Implement

ADDIE
Evaluate



- Good designs
- Architecture
- Technical excellence
- Simplicity
- Changing requirements
- Working together
- Motivated individuals
- Face to face conversation

- Good designs
- Architecture
- Technical excellence
- Continuous delivery
- Become more effective

- Good designs
- Architecture
- Technical excellence
- Deliver frequently
- Continuous delivery
- Become more effective

- Good designs
- Architecture
- Technical excellence
- Deliver frequently
- Continuous delivery
- Become more effective

- Good designs
- Architecture
- Technical excellence
- Working software
- Satisfy the customer

Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Solutions: How We Get There in Agile Metrics

ADDIE
Develop

Are you meeting commitments?

Project tracking

What is your current pace?

ADDIE
Develop

How much code is getting built?

Source control

How well is the team working together?

ADDIE
Develop

How long does it take you to get things right?

Continuous integration

ADDIE
Implement

How fast can you get changes to your consumers?

Deployment tools

ADDIE
Evaluate

How well is your system performing?

Application monitoring

How are your customers using your system?

Bottom left corner of each slide that follows indicates an approved capability requirements package passed to the DoD ADL that can fully or substantially mitigate those gaps.

Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Solutions: Agile Metrics for Develop (Project Tracking 1 of 2)

ADDIE
Develop

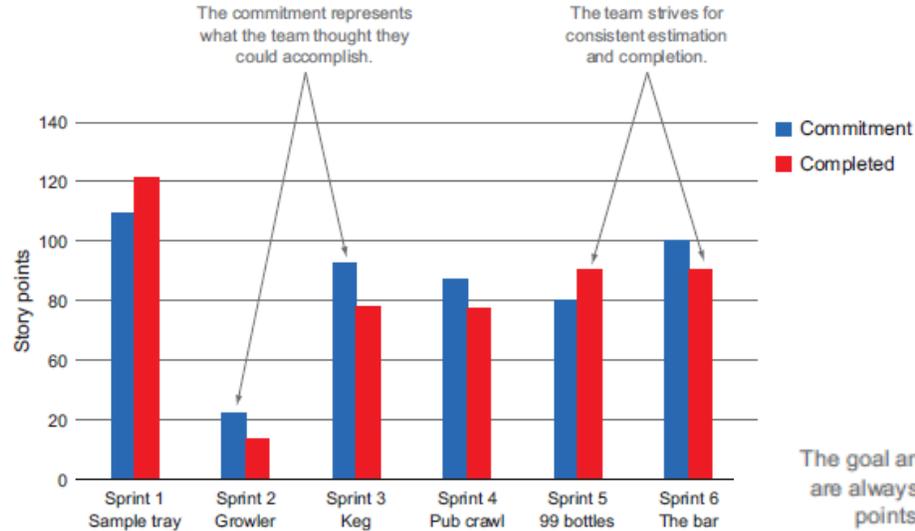
Are you meeting commitments?

Project tracking

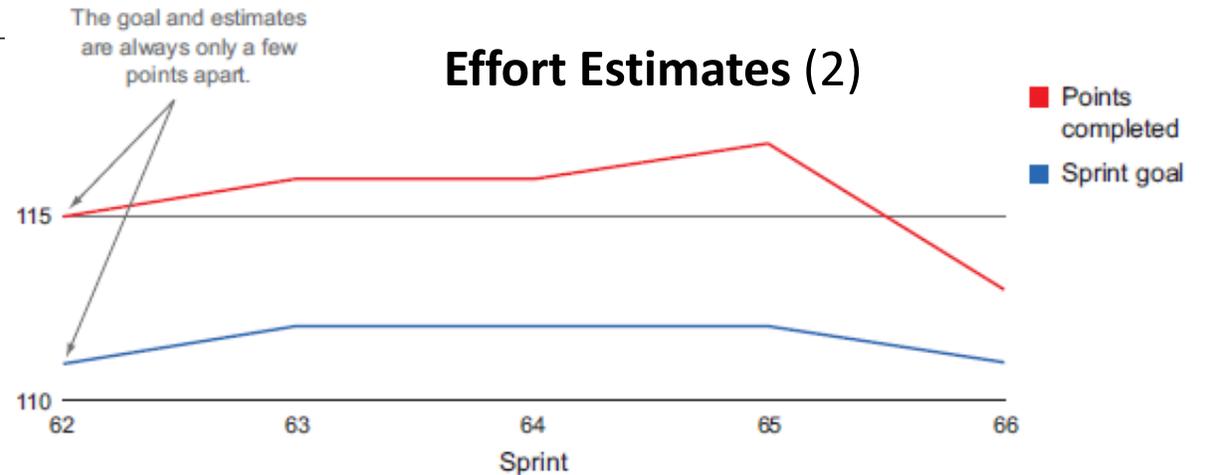
What is your current pace?

DL Registry (DLR)

- 105 requirements approved 2013



Velocity (1)



Effort Estimates (2)

BL –The two “most influential” factory metrics” are project tracking metrics.

Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Solutions: Agile Metrics for Develop (Project Tracking 2 of 2)

ADDIE
Develop

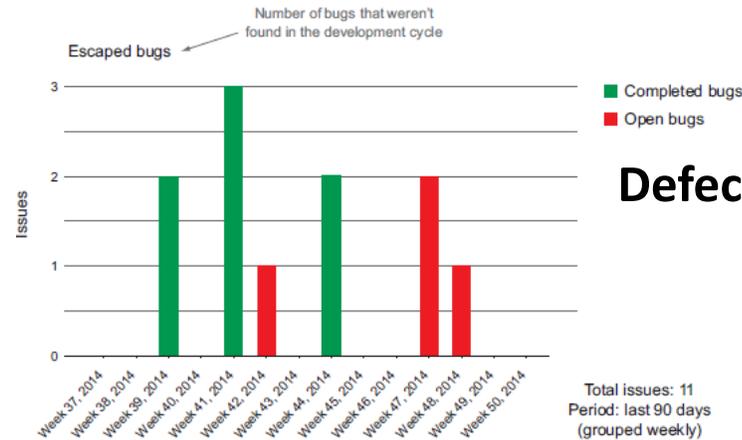
Are you meeting commitments?

Project tracking

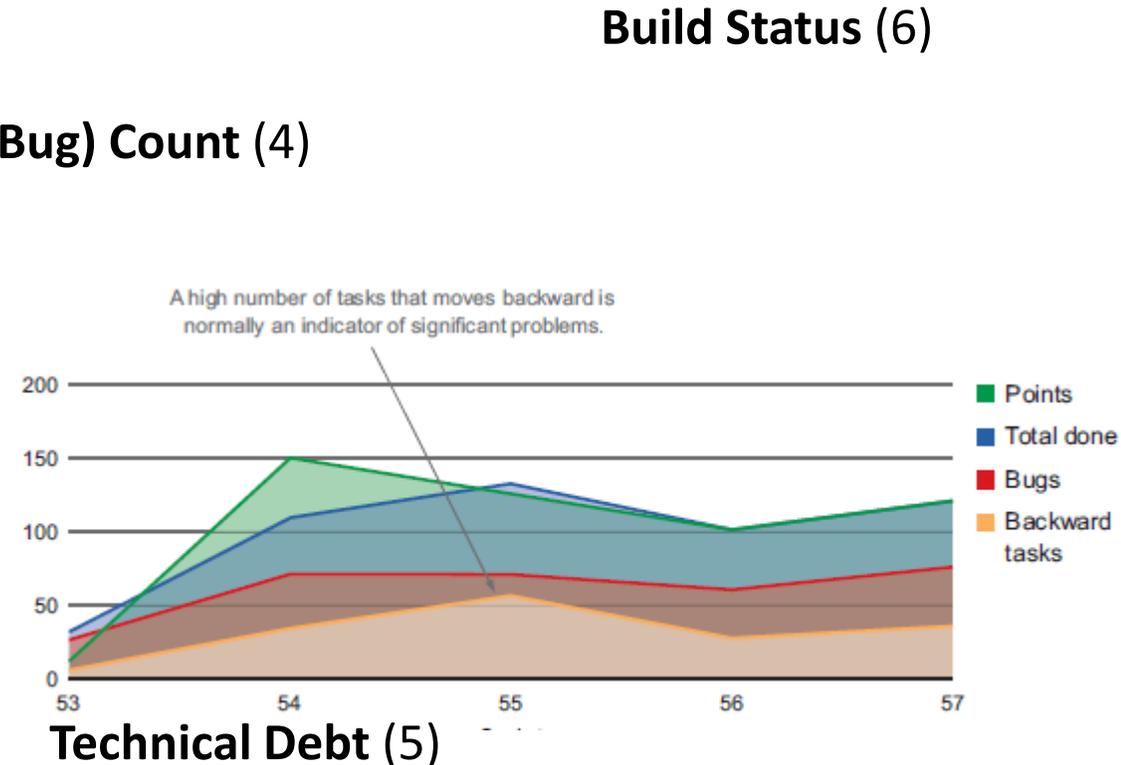
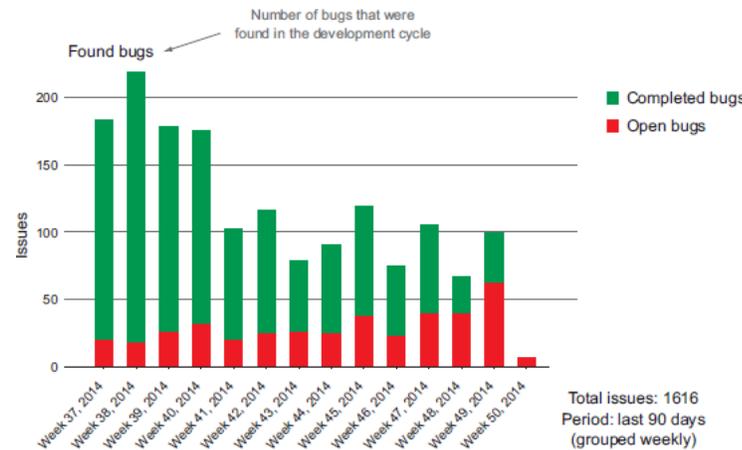
What is your current pace?

DL Registry (DLR)

- 105 requirements approved 2013



Defect (Bug) Count (4)



Technical Debt (5)

BL – 5 of 6 of the “most influential” factory metrics” are project tracking metrics.

Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Solutions: Agile Metrics for Develop (Source Control)

ADDIE
Develop

How much code is getting built?

Source control

How well is the team working together?

DL Source File Repository (DLSFR)
- 41 requirements approved 2015

Pull Requests

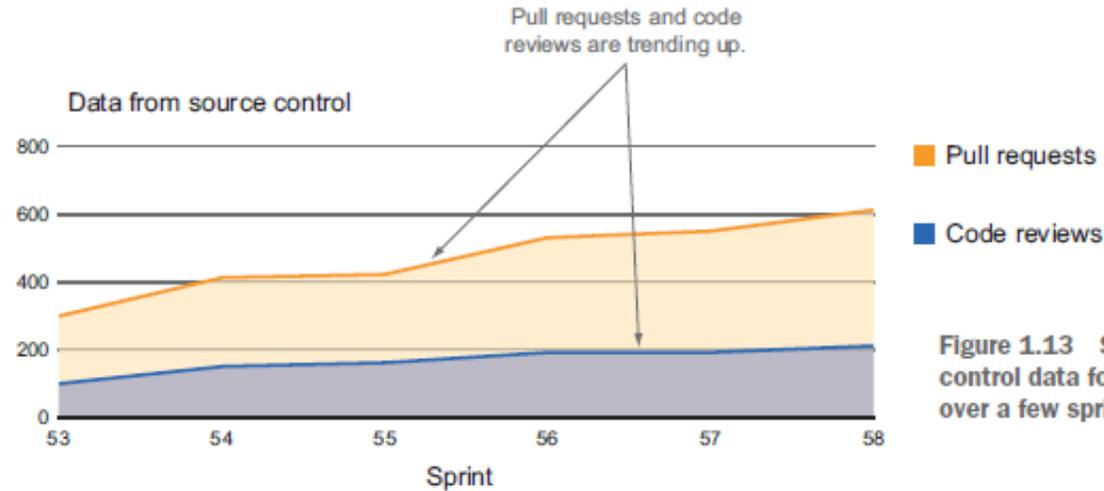
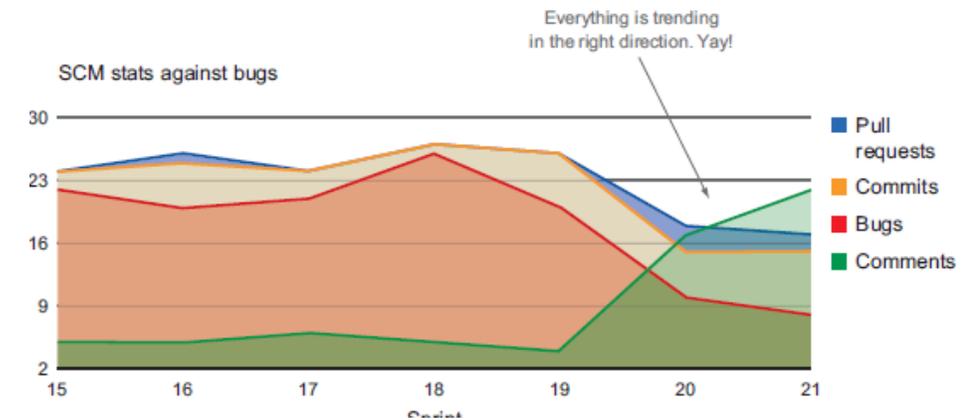


Figure 1.13 Source control data for a team over a few sprints

Commits, Reviews, Comments, CLOCs



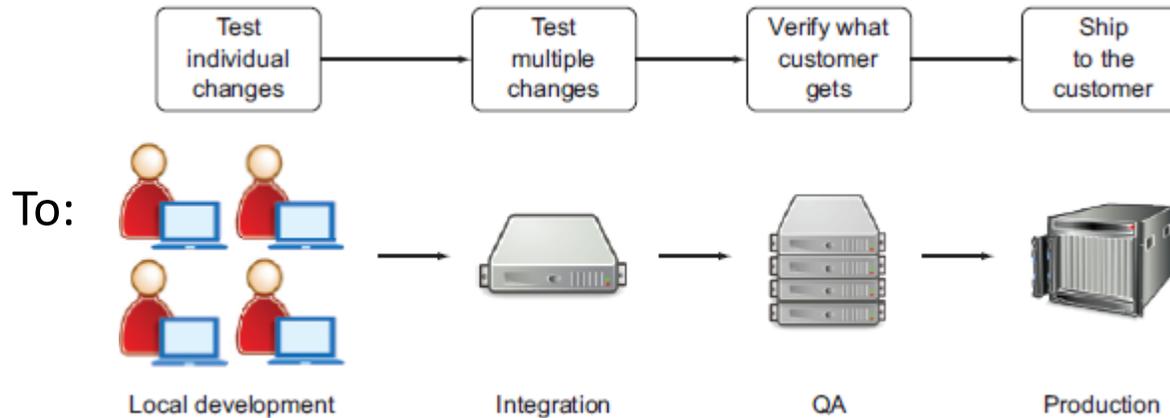
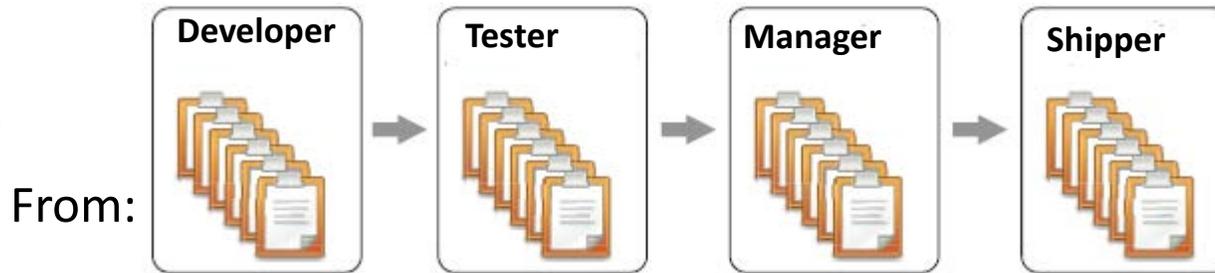
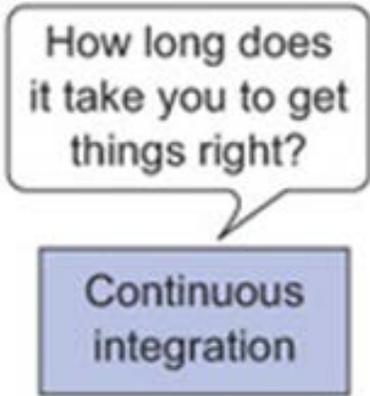
Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Solutions: Agile Metrics for Develop (Continuous Integration)

ADDIE
Develop

Progress as working code (7)

Manual
Content (Doctrine) Validation
Section 508 Testing
Presentation Code Testing
Scoring Language Code Testing
Individual Trials (Iterative)
Group Trials (Iterative)



*DLICR also auto detects and stores learner computing environment data.
DL Issue Collection Repository (DLICR) - 93 requirements approved 2015*

Task ID	Task Name	Start	End	Duration	Dependencies	Notes
14	14	14	14	14 days	As Soon As Possible	Fixed Duration
15	15	15	15	15 hrs	As Soon As Possible	Fixed Duration
16	16	16	16	16 hrs	As Soon As Possible	Fixed Duration
17	17	17	17	17 hrs	As Soon As Possible	Fixed Duration
18	18	18	18	18 hrs	As Soon As Possible	Fixed Duration
19	19	19	19	19 hrs	As Soon As Possible	Fixed Duration
20	20	20	20	20 hrs	As Soon As Possible	Fixed Duration
21	21	21	21	21 hrs	As Soon As Possible	Fixed Duration
22	22	22	22	22 hrs	As Soon As Possible	Fixed Duration
23	23	23	23	23 hrs	As Soon As Possible	Fixed Duration
24	24	24	24	24 hrs	As Soon As Possible	Fixed Duration
25	25	25	25	25 hrs	As Soon As Possible	Fixed Duration
26	26	26	26	26 hrs	As Soon As Possible	Fixed Duration

Because Individual and Group Trials require human learners take them for ISD purposes of establishing course length and item analysis difficulty measurement IT and GT may never be fully automated..

Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Solutions: Agile Metrics for Implement (Deployment Tools)

ADDIE
Implement

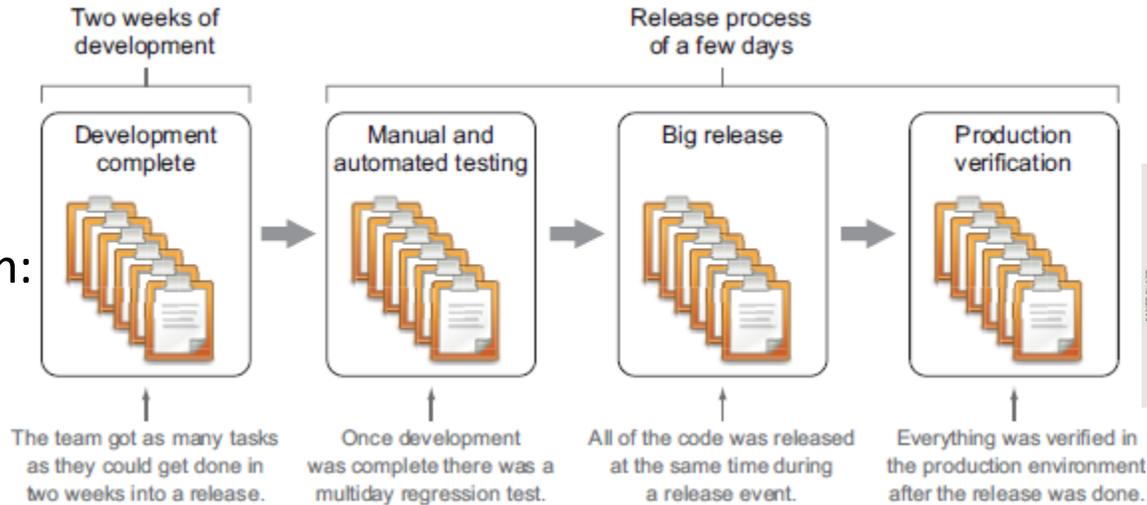
Progress as working code (7)

Manual
Documents/Test Logs Review
Function Testing
Fielding

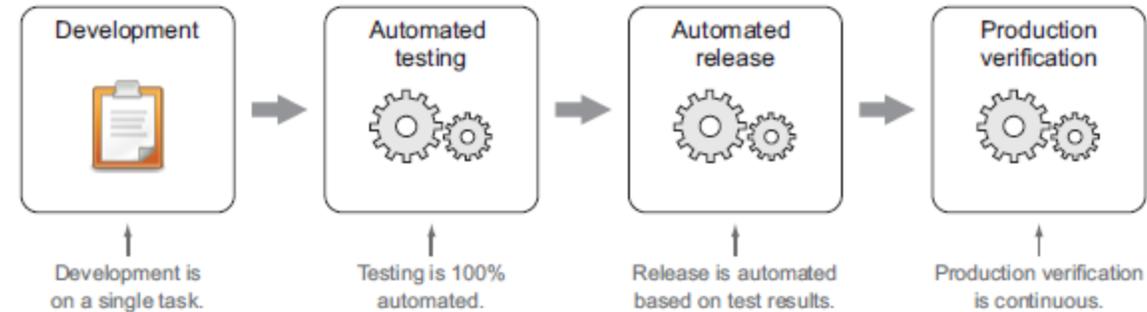
How fast can you get changes to your consumers?

Deployment tools

From:



To:



Task ID	Task Name	Start	End	Duration	Dependencies	Notes
26	# ADDIE IMPLEMENT: Stage (GAB and FT) (ex. In 48 hours of level 2 CMBM)	19:38	19:38	0 days	As Soon As Possible	Fixed Duration
27	# 5a. Prepare for A Conduct Gov't Acceptance Review (GAR)	19:25	19:25	0 days	As Soon As Possible	Fixed Duration
28	Log receipt of deliverables	20	20	0 hrs	Start to Earlier Than	Fixed Duration
29	Review deliverables for completeness	20:32	20:32	0 hrs	As Soon As Possible	Fixed Duration
30	Assess for GAB	31	31	1 hr	As Soon As Possible	Fixed Duration
31	Conduct Log Review	66	66	66 hrs	As Soon As Possible	Fixed Duration
32	Conduct screenshot review: 26 x (1 * 5MCHURS)	51	51	0 hrs	As Soon As Possible	Fixed Duration
33	Generate report of findings	54	54	0 hrs	As Soon As Possible	Fixed Duration
34	# Prepare GAB (and update DL Registry with results)	56	56	1 hr	As Soon As Possible	Fixed Duration
35	# 5b. Prepare for and Conduct Function Test (FT)	88:38	88:38	0 days	As Soon As Possible	Fixed Duration
36	Transfer files to Delivery System	31	31	0 hrs	As Soon As Possible	Fixed Duration
37	Conduct deliverable QA	38	38	0 hrs	As Soon As Possible	Fixed Duration
38	Load files into repository	26	26	0 hrs	As Soon As Possible	Fixed Duration
39	Build course structure: 19 x (3.6 * 5MCHURS)	46	46	0 hrs	As Soon As Possible	Fixed Duration
40	Conduct FT: 48 x (1 * 5MCHURS)	41	41	0 hrs	As Soon As Possible	Fixed Duration
41	Generate report of findings	42	42	0 hrs	As Soon As Possible	Fixed Duration
42	5a. Complete Function Test (and update DL Registry with results)	45:47:48	45:47:48	1 hr	As Soon As Possible	Fixed Duration
43	# Pending Activities (1-18 to 12 based hours before post-reading Operational Tryout)	4	4	0 days	As Soon As Possible	Fixed Duration
44	# 5c. # Complete Level 2.4 review?	1	1	0 day	As Soon As Possible	Fixed Duration
45	# 5d. # ATRIS Management?	1	1	0 day	As Soon As Possible	Fixed Duration
46	# 5f. Hold DL on LMS	2	2	0 days	As Soon As Possible	Fixed Duration

*DLICR also auto detects and stores learner computing environment data.

DL Delivery Systems (ETS, LMS, LCMS for CMI; CMS for CAI; Etc.) All are deployed now.

Automated



Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Way-Ahead: Require ADDIE Develop Stage Metrics

ADDIE
Develop

Are you meeting commitments?

Project tracking

What is your current pace?

DL Registry (DLR)
- 105 requirements approved 2013

ADDIE
Develop

How much code is getting built?

Source control

How well is the team working together?

DL Source File Repository (DLSFR)
- 41 requirements approved 2015

ADDIE
Develop

How long does it take you to get things right?

Continuous integration

*DLICR requirement set has a sub req set for automatically detecting and storing learner's computing environment metrics.

DL Issue Collection Repository (DLICR)
- 93 requirements approved 2015*

ADDIE
Implement

How fast can you get changes to your consumers?

Deployment tools

DL Delivery Systems (ETS, LMS, LCMS for CMI; CMS for CAI; Etc.)
All are deployed now.

ADDIE
Evaluate

How well is your system performing?

Application monitoring

How are your customers using your system?

DLICR (help desk); ETS, LMS, LCMS must support item analysis
CMI Interaction data

Davis, C. W. H. (2015). Agile metrics in action : how to measure and improve team performance

Conclusion

BL - Much progress in DL metrics analytics has been but more is needed.

Q & A

- Contact mitchell.l.bonnett.civ@mail.mil or mitch_bonett@hotmail.com