Federal Sector Agile Productivity Case Study

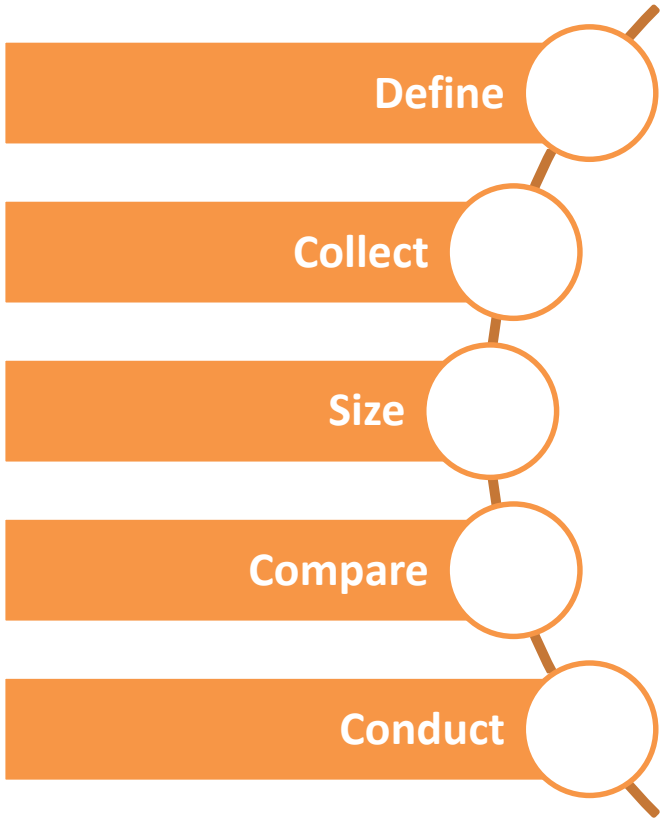
Prepared for:
2019 ICEAA Conference

Presented By:
Kevin McKeel

LOGAPPS

May 11, 2019

Agenda



- Background, characteristics, and principles of Agile Usage at the IRS
- Program documents to include Requirements and User Stories
- Each program based on IFPUG functional size measurement processes and elaboration meetings
- Productivity between Agile projects and Waterfall projects, and implement benchmarking program
- Project Management interviews to identify best practices and challenges

Characteristics of Agile

Vision and customer value driven

- User requirements change over time
- User requirements follow the cone of uncertainty
- Responding to change is critical

Iterative, feature driven development

- Delivery every cycle (1 – 4 weeks)
- Full lifecycle duration
- Cycle ends with a user review (demo)
- A release plan outlines product development

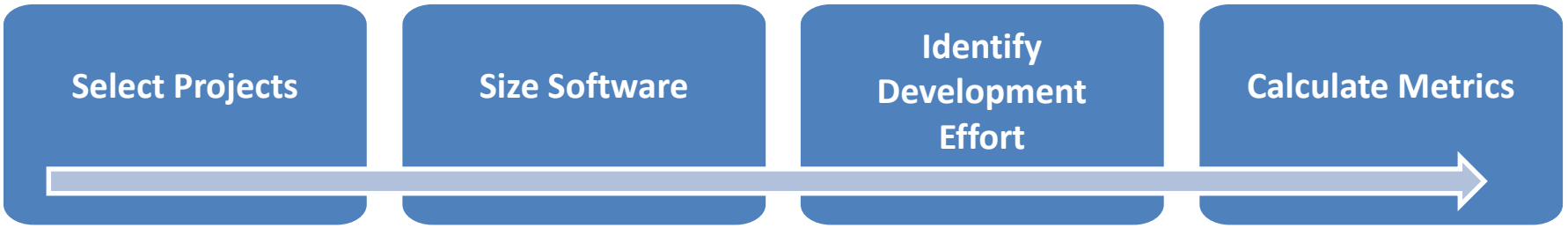
Collaborative

- Team intimacy (product owner and development staff)
- Short, shorter, shortest feedback loops
- Self organizing and self managing

Agile Adoption within the IRS

- IRS adopted Agile development process and invested in training
- Agile projects at IRS (and in federal government generally) are not 'traditional agile' due to acquisition and budgeting constraints
- Many IRS projects have annual deployments aligned to the tax calendar
  - while industry use of Agile typically involves continuous delivery
- Agile principles at the IRS typically include :
  - scrum teams
  - time-boxed sprints
  - documenting requirements in the form of user stories
- Projects were completed between FY16-FY18

Analysis Process



- Projects include fraud detection systems, public facing web services, tax databases, and tax registration systems
- Projects selected on basis of Agile delivery where software requirements or User Stories were available

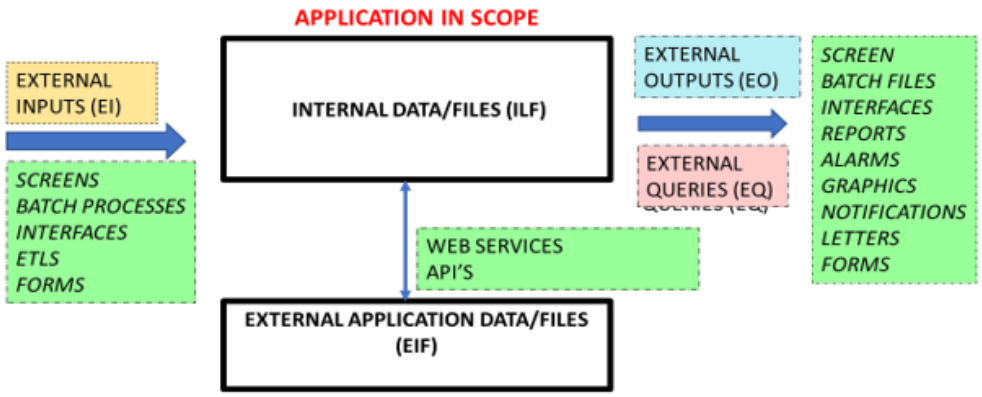
- Projects include fraud detection systems, public facing web services, tax databases, and tax registration systems
- Projects selected on basis of Agile delivery where software requirements or User Stories were available

- Projects include fraud detection systems, public facing web services, tax databases, and tax registration systems
- Projects selected on basis of Agile delivery where software requirements or User Stories were available

- Projects include fraud detection systems, public facing web services, tax databases, and tax registration systems
- Projects selected on basis of Agile delivery where software requirements or User Stories were available

Functional Size Assessment Process

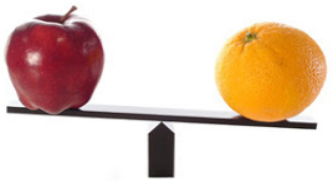
- Analyze system architecture to determine the features in scope and the boundaries
- For each application within the scope of the project
  - **Count data stores** which are maintair used, or referenced as data functions (ILF, EIF)
  - **Count data flows** which are input, output, and inquiry transactions as transactional functions (EI, EO, EQ)
  - **Assign a complexity** for each functior low, average, or high based upon dat usage/data flow rules
  - **Assign a value** to each function based upon its complexity (range = 3-15 fps depending on type)
  - **Sum the values** of all functions for the application project count



LOGAPPS

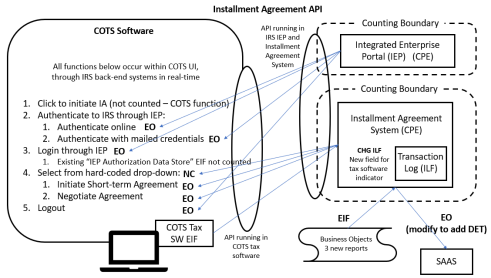
How does Functional Size Estimation differ with Agile?

- Certified Function Point Specialist provided the functional sizing for all projects, which is a software estimation best-practice
- Primary difference between Agile and Waterfall is the timing and detail of system requirements
  - In Agile, the backlog and lean documentation are provided, with high-level details
  - In Waterfall, detailed business system requirements are documented to include functional requirements, non-functional requirements, and interface control documents
  - Primary difference is understanding complexity of functions in Agile, not the existence of functionality
- We overcame this obstacle by applying risk and through the assumption that all functions are of average size, which is also consistent with Simple Function Points



Functional Scope Elaboration

- Requirements elaboration sessions with Agile teams are a useful method to identify additional features that have not been documented
- Functional sizing expert can 'white board' system boundary, define interfaces, and diagram data flow by interviewing Product Owner or System Design SMEs
- Scope elaboration meetings also increase confidence in technical SMEs of estimation process
- Often results increase complexity of specific functions (additional FTRs)
- Requires ability to translate technical implementation into functional scope
- Seasoned estimator can help author user stories as functionality is defined



LOGAPPS

Budget v Estimation v Planning

Budgeting

- Defines how much we have to spend and influences scope
- Tends to ignore the cone of uncertainty

Estimation

Rough or approximate size extent or nature

- Focused by the cone of uncertainty, ranges

Planning

Definition of tasks and allocation of resources

- Focused on the narrow part of the cone of uncertainty

Benchmarking

- Best practice in software estimation is to benchmark projects within own development organization by vendor and technology
- Develop local productivity factors from language, software size, and staff-months
- Ensure project financial data has been verified
- Use Productivity factors as primary methodology or as crosscheck
- Collect qualitative information

Delivered Software Size (Physical SLOC):	175,371 Java (new)
Delivered Software Size (Physical SLOC):	8,760 Java (reused)
Logical SLOC:	202,370
Effective SLOC Size:	196,240
Language(s):	Java, DB2
Average/Peak Staff:	27/75
Staff-months (MS2-MS4b):	946
Effective SLOC Productivity:	208 SLOC/PM
AD Labor Mix - % In-House:	54%
Deployment staff as percent of peak:	36%

Project Type: Electronic Data Interchange, Database Development
Acquisition Method: New Development
Platform: Server (Sun)
ELC Path: Iterative/Agile
Funding Classification: DME
Cost Completeness: Partial
Domain: Submission Processing
Owner: LB&I
Manager: John Smith AD:SP
Project Summary Last Updated: 9/8/2014

ROM-Level Function Point Analysis

Fast Function Points

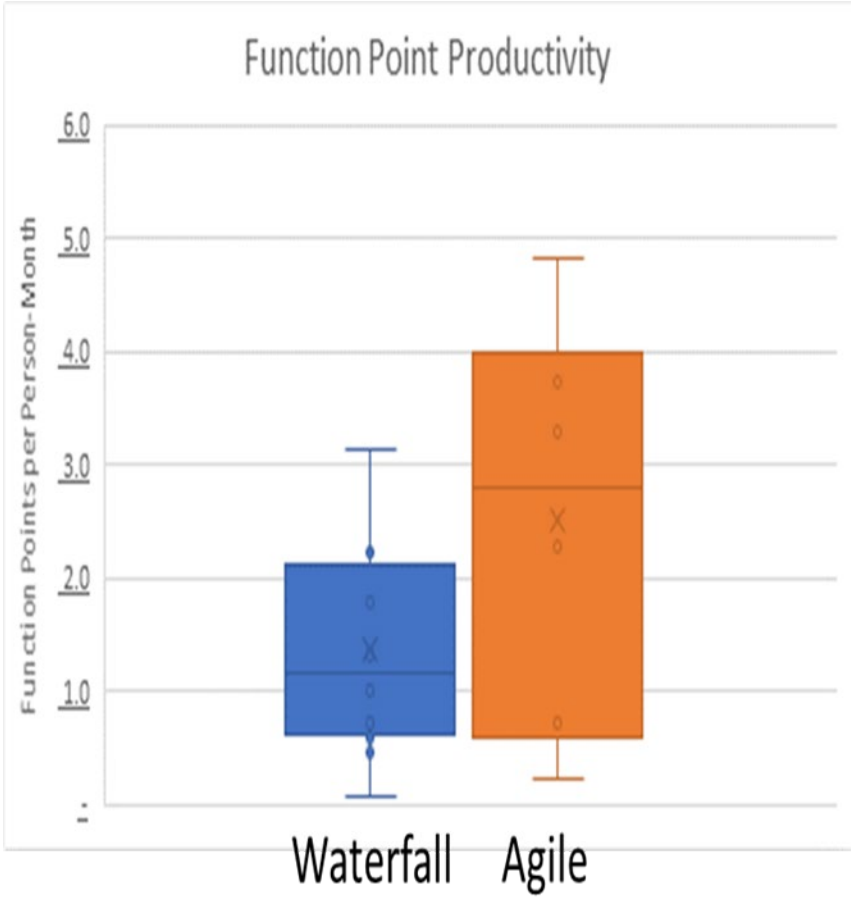
- Count data and transactions in accordance with IFPUG rules
- Ignore DETs and RETs (not yet identified)
- Apply average weightings unless complexity can be interpreted
- Difficult to separate Outputs and Queries; data groups are not always identifiable as external or internal

Simple Function Points

- Identify Unspecified Generic Data Groups (UGDP)
- Identify Unspecified Generic Elementary Process (UGEP)
- Does not differentiate Internal or External Data storage; does not differentiate inputs, outputs, or inquires
- Apply weights (UGDP=7.0, UGEP =4.6)

Developer Productivity Change – Measured in Function Points

- The IRS database of Waterfall projects has a median productivity of 1.2 function points per person-month
- Agile projects have a median productivity of 2.8 functions points per person-month
- IRS Projects, in general, tend to have lower productivity due to high testing and security levels, heavy system integration, and adherence to tax calendar



Summary Results

Project	Web Portal	Foreign Entity Reporting	Tax Registration	Data Warehouse I/F	Financial Reporting	Fraud Detection
Function Points	3,785	957	1,439	255	148	838
Person Months	1,649	328	614	355	661	254
FP/Person Month	2.30	2.92	2.35	0.72	0.22	3.30
Schedule	46	11	18	25	24	9
Language	Java	Java	Java	Informatica (XML)	Junit, Java, Drools	Business Objects

- Web Portal is inclusive of all software delivered through August 2018, thus has largest size and effort
- Data Warehouse and Financial Reporting projects schedules are not indicative of an Agile project, and are considered outliers
- Assuming a backfire ratio of 53, the median value of 2.80 equates to 149 SLOC/person-month. IRS projects, as tracked by IRS Estimation Program Office, have a median SLOC per person-month of 104 (35% below Agile projects)

IRS Program Manager Observations

IRS IT Project Managers were interviewed, and made numerous observations that impact benefits of Agile

- Development teams are 60% federal, 40% contractors, and government assumes system integration role
- While programs may be Agile, the delivery partners are Waterfall
- Time delays for environment construction are common
- IRS, like many federal agencies, have processes that are difficult to overcome simply by changing software development methodology
- Budget cuts have forced reduction in contractor staff, and federalization of Application Developers, and likely loss of institutional learning
- Agile is more effective with projects not influenced by tax calendar, such as Web Portal
- Contracting process is an ongoing challenge
- Projects invest in Non-Functional user stories, including analysis, spikes, technical implementation, and environment setup

What does this mean to the Agile Software Estimator?

- Identify purpose (planning, budgeting, or estimation)
- Identify the flavor of Agile (Scrum, XP, Kanban), experience with Agile, and enterprise adoption
- Ensure estimation life-cycle is aligned to known epics, features, or user stories
- Use Simple Function Points to size known features or User Stories
- Incorporate functional size elaboration sessions with Agile team
- When using parametric models, calibrate to local productivity factors by implementing benchmarking process
- Avoid analogy estimation due to differences across Agile teams and terminology
- Identify functional and non-functional requirements and account for non-functional Agile Release Trains

CONCLUSIONS

Conclusions

- The federal government has been adopting Agile for almost ten years. **AGILE IS HERE TO STAY**
- General agreement that productivity has improved using Agile
  - This study alone found 35% gain
- Projects need strong support from executive layer to counter bureaucratic challenges
- Functional scope elaboration meetings should be part of sizing process
- Invest in benchmarking processes

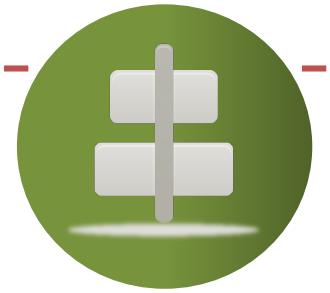
MEETING THE CHALLENGES POSED BY NEW PROCESSES

? How are we delivering our products?

? What is the appropriate size and frequency?

- Changes in the way our customers behave means we must change what services we provide and how we provide them

Product Delivery



ALIGN

Align Products to Release Plans and Milestones

CONDENSE

Condense Deliverables

- ✓ Remove repetitive documentation
- ✓ Executive-level briefs (GAO style)

AGILE REQUIRES CHANGES IN COST ESTIMATION SERVICES

- More frequent estimation support outside of budgeting cycle
- More communication with stakeholders and executive sponsors
- More emphasis on upcoming releases rather than full lifecycle estimate
- More focus on estimating the capability that can be delivered within a given budget
- More focus on technical debt in operations and maintenance (O&M) phase of lifecycle.

Faster, Agile delivery requires faster, more agile, analysis