
Agile for Information Development

August 1, 2012

Laura Clymer



Agenda

- Introduction
- Agile
 - Definition
 - Base Model
- Existing Info Dev Model
- Experiences on Recent Programs
- What Did We Learn
- Changes for the Future
- Conclusion, Q&A

Introduction - Laura Clymer

Information Development professional - 16 years of experience

- Semiconductor hardware, software products
- Implemented 3 ground-up conversions to SGML/XML
- Global authoring considerations - writing/SME spanned the global

Information Development Senior Manager with Dell

- Team of 6 writers, 1 illustrator
- Software products
- Using DITA XML as source content for over a year

Agile Definition

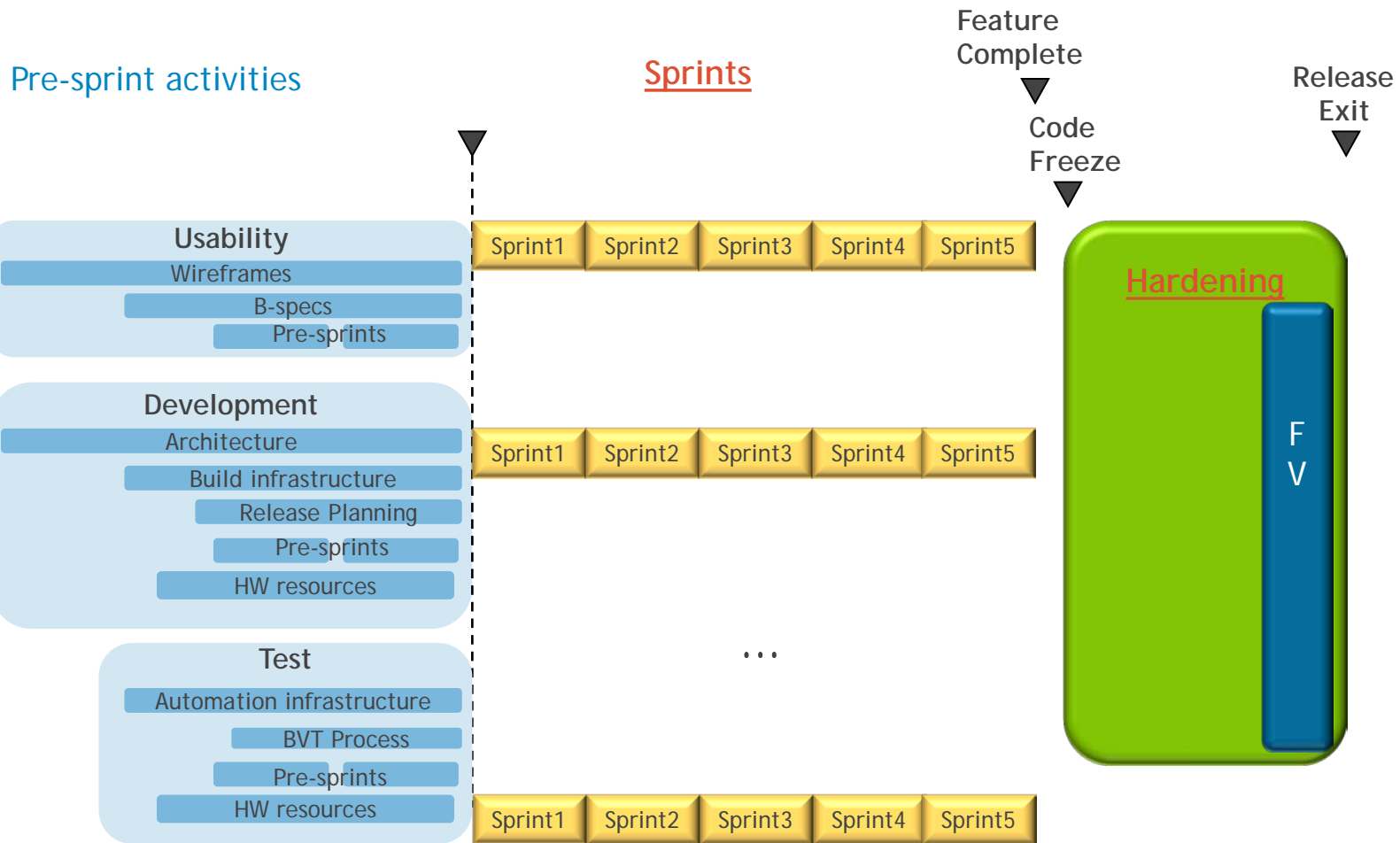
Agile software development is:

- Group of software development methodologies
- Based on iterative and incremental development
- Solutions evolve through collaboration of cross-functional teams

Agile Manifesto (2001)



Base Development Model - Agile



PRP

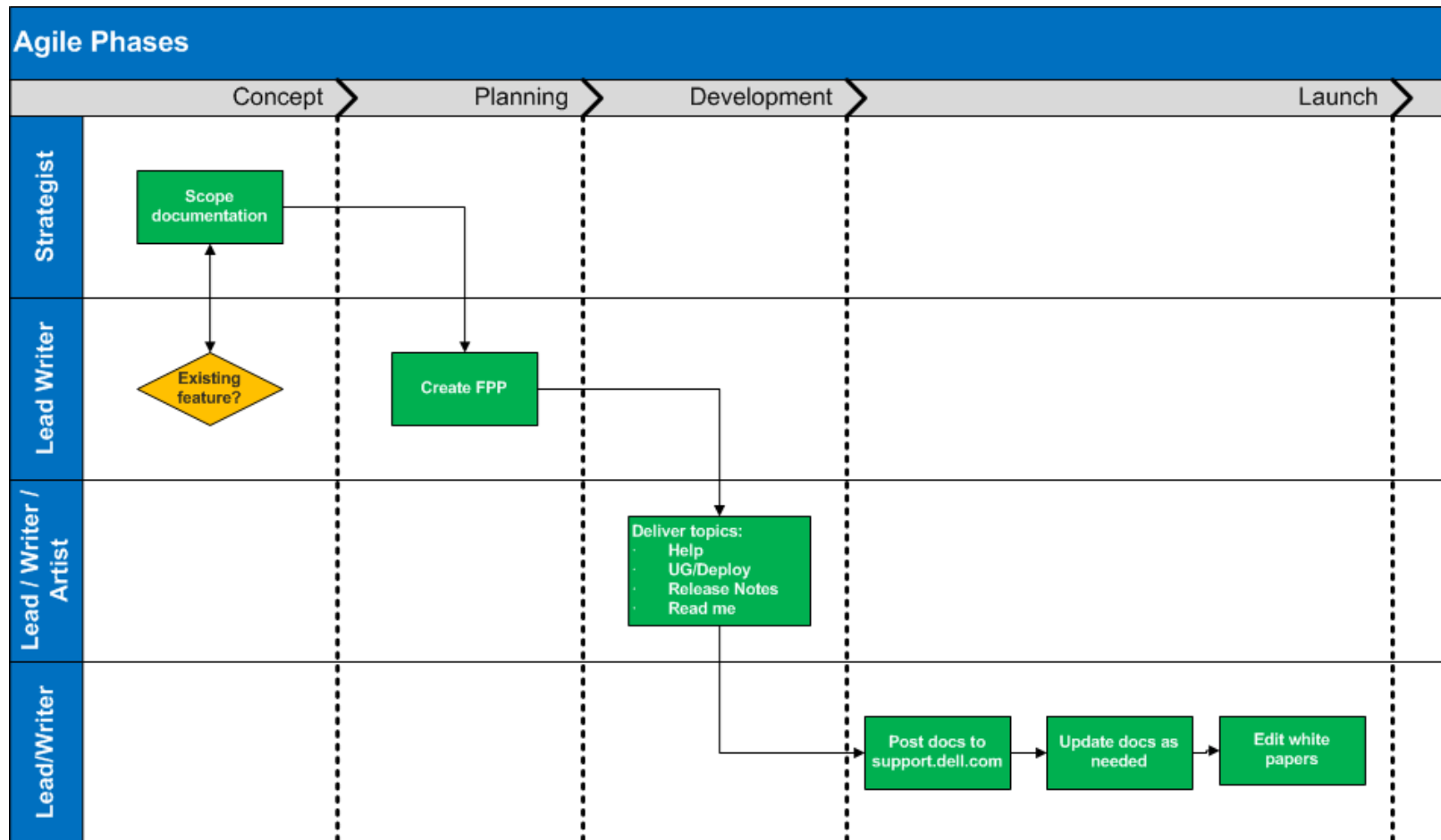
Define

Plan

Develop

Launch

Previous Info Dev Lifecycle



Info Dev stayed with traditional delivery model, even though Dev changed!!

Experiences on Recent Programs

Program	Continue	Lessons
<u>Program 1.0</u>	<ul style="list-style-type: none">• Attended Release Planning Sessions• Attended Scrum meetings• Delivered help files to development team during sprints	<ul style="list-style-type: none">• Functional Information Development Plan started too late, too little detail• Documents are not inputs to test• Wanted full guides for review• No acceptance criteria
<u>Programa</u>	<ul style="list-style-type: none">• Attended Planning Sessions• Attended Scrum meetings• Delivered GUI text reviews during sprints	<ul style="list-style-type: none">• Error messages written using different standards - no consistency• Content was reviewed eight times (4 help, 4 guide)• Understaffed
<u>Ghost</u>	<ul style="list-style-type: none">• Attended Release Planning Sessions• Attended Scrum meetings• Delivered help files to development team during sprints	<ul style="list-style-type: none">• User interface text rewritten without Info Dev input/review• Error messages not written until final validation• Development team switched back and forth between Agile and Waterfall

What Did We Learn?

Functional Information
Development Plan

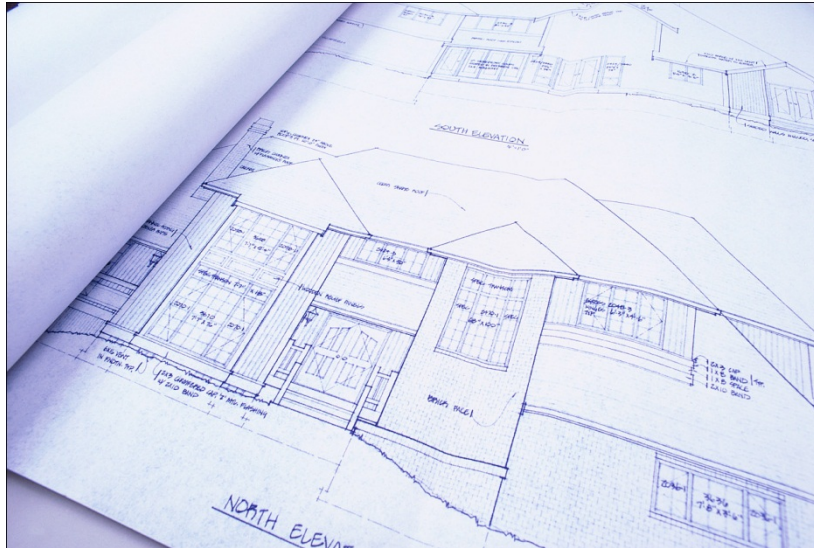
Review Cycle

Phase Appropriate Interlocks

Other Findings

Functional Information Development Plan

- Start early to include scoping/requirements work



- Contains high-level information including deliverable descriptions, dates, timelines, and language requirements

Review Cycle

- Deliver iteratively for review - no more full books
- Once reviewed/approved, not sent through again
- Topic content is reused between help and guides
- One full end-to-end review of guides



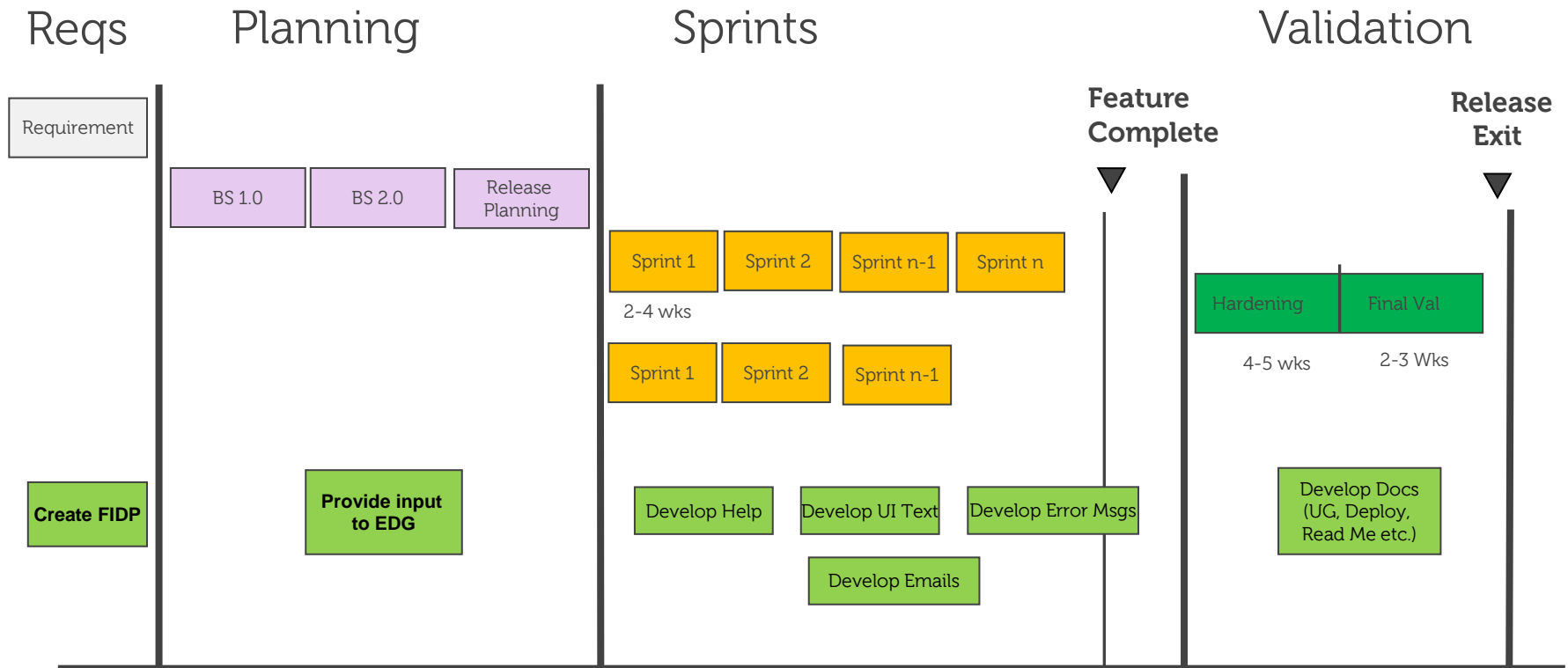
Other Findings

- Use guidelines for consistency
- Acceptance criteria must be in place for our deliverables
- GUI text must have edit
- Increase staffing ratios
- New authoring system





Agile+Info Dev Phases



NOTE

- All Info Dev deliverables must be determined prior to Sprint 1, including Help, Guides, UI Text, System-Generated Emails, Error Messages, Read Me etc.
- Info Dev should touch ALL customer-facing text.
- Info Dev will deliver topic-based documentation on a sprint-by-sprint basis.

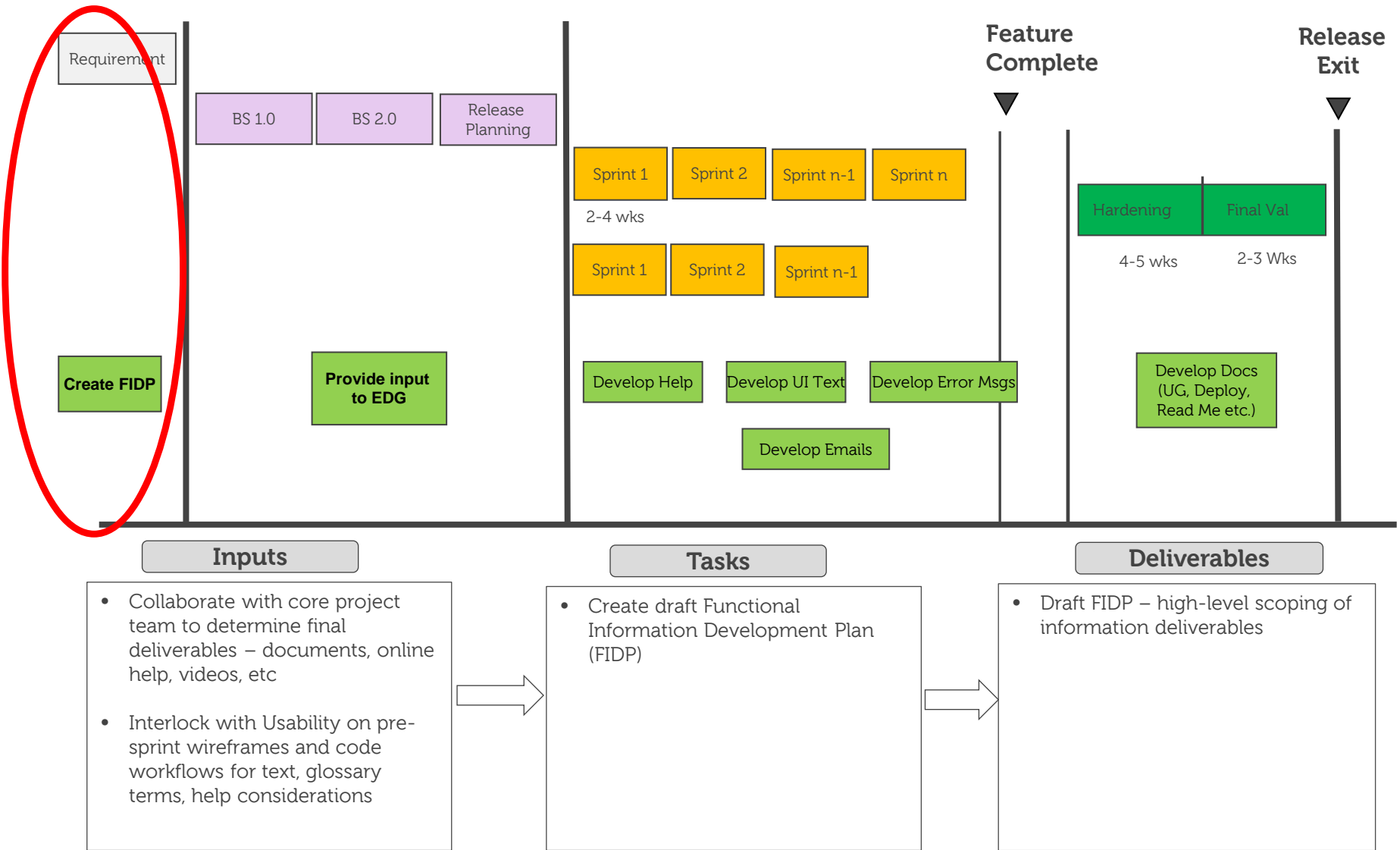
Requirements Gathering

Reqs

Planning

Sprints

Validation



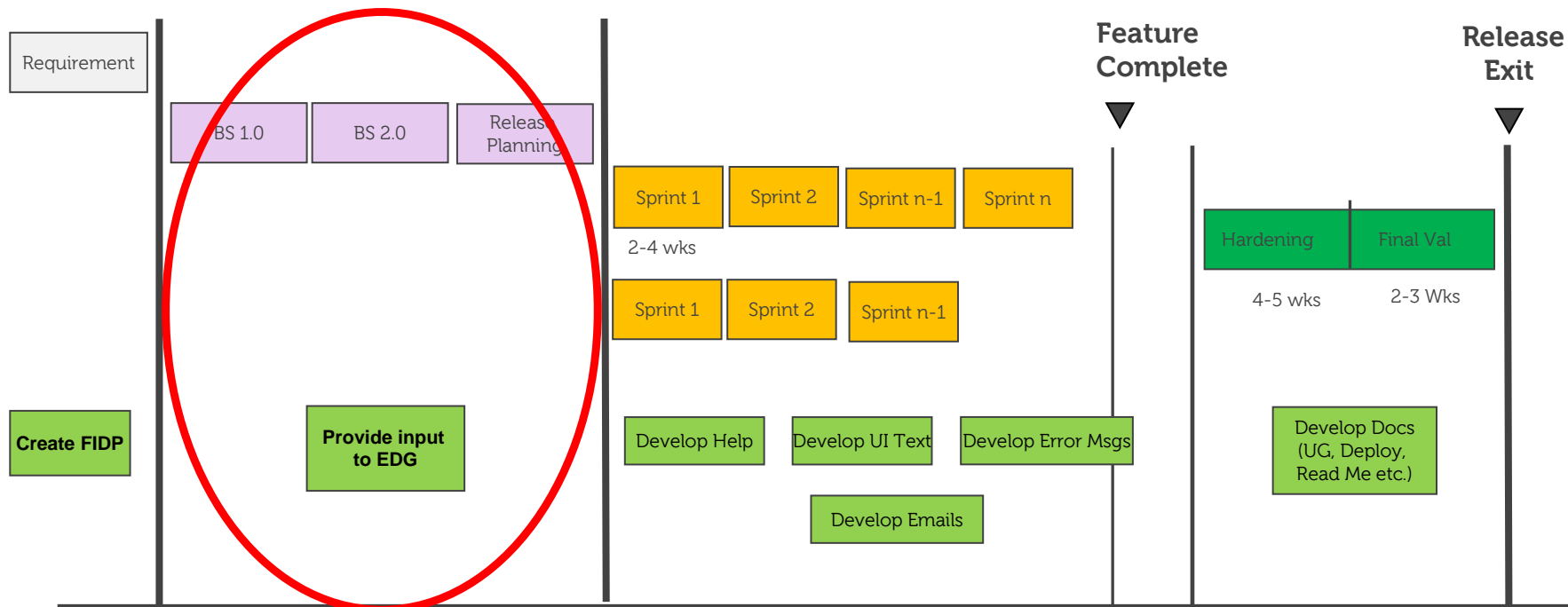
Planning

Reqs

Planning

Sprints

Validation



Inputs

- Provide text and help input to EDG on UI content design
- Provide error message guidelines, review considerations, information scoping to scrum teams

Tasks

- Analyze UI design
- Understand usability testing plan
- Enter acceptance criteria for error messages, help topics, other code-based deliverables
- Final FIDP

Deliverables

- Feedback on UI content requirements
- Final FIDP – this is Info Dev's committed plan that we deliver against
- Acceptance criteria

Key Planning Considerations:

- **Functional Information Development Plan**
 - Contains high-level information including deliverable descriptions, dates, timelines, and language requirements
 - Documents are **not** required inputs to test – documents are created/finalized after various test cycles are complete
- **Review Cycle**
 - Deliver topics iteratively to team for review during sprints – full books are no longer delivered until hardening
 - Once topic is reviewed/approved, not sent through review again until hardening
 - Topic content is reused between help and guides
 - One full end-to-end review of Guide content during hardening

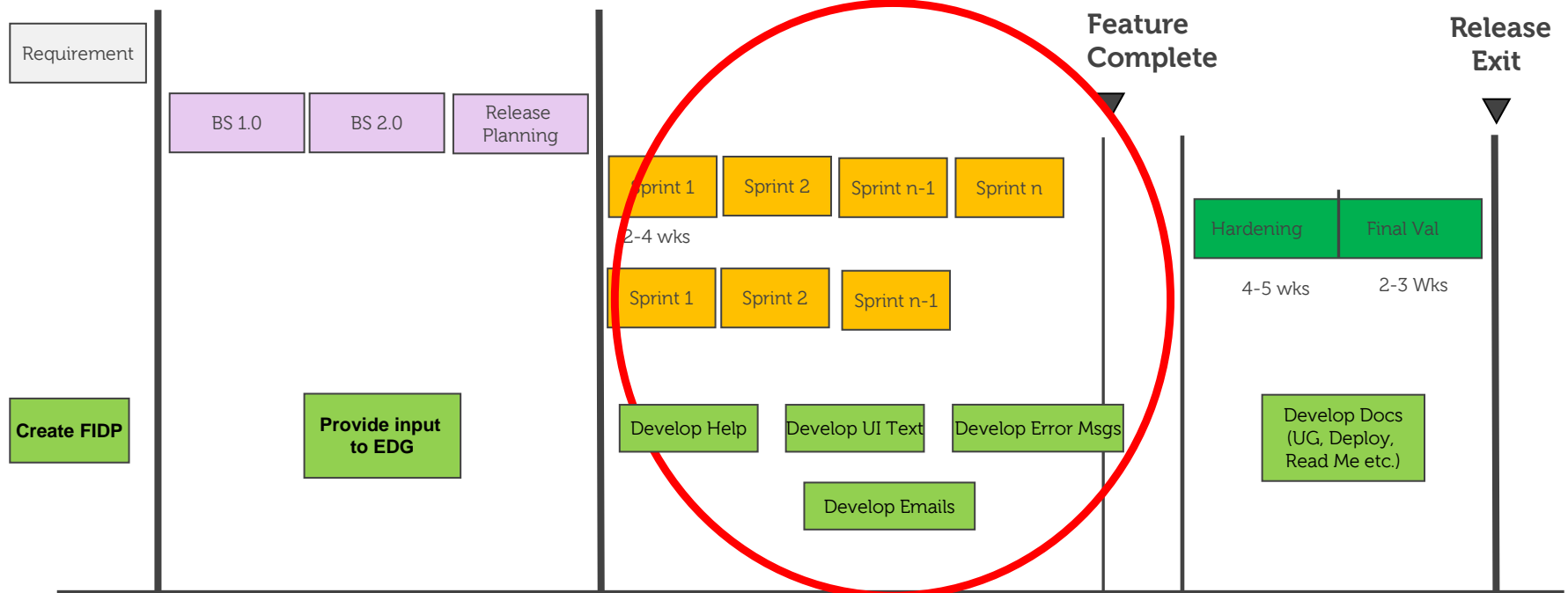
Sprint

Reqs

Planning

Sprints

Validation



Inputs

- Participate in sprint planning meetings for user story & task identification
- Review TFS/Rally/MRD for documentation impacts, updates, additions/deferred features

Tasks

- Develop DITA Topics
- Manage topic reviews
- Update topics per Dev/Test feedback

Deliverables

- Integrated Help System
- Edited UI Text
- Edited Error Messages
- System-Generated Emails
- Other deliverables as required

Key Sprint Considerations:

- **Review Cycle**

- Deliver topics iteratively to team for review during sprints – full books are no longer delivered until hardening
- Once topic is reviewed/approved, not sent through review again until hardening – part of story acceptance criteria
- Topic content is reused between help and guides
- One full end-to-end review of Guide content during hardening

- **Error Messages**

- Delivered with code utilizing Error Message Guidelines
- Dev sends message to Writer mid-sprint, and Writer returns to Dev w/ changes
- Part of feature complete – part of story acceptance criteria

- **GUI Text**

- All text must be edited by Info Dev prior to feature complete – part of story acceptance criteria

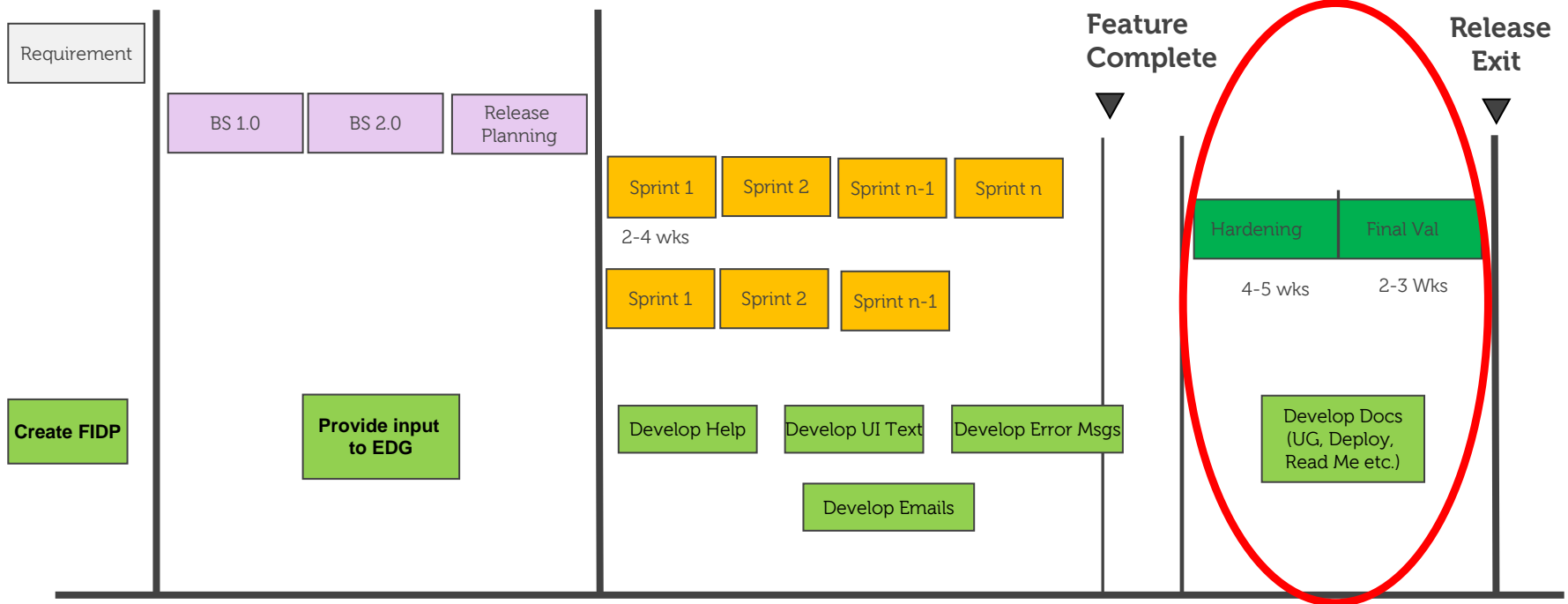
Testing

Reqs

Planning

Sprints

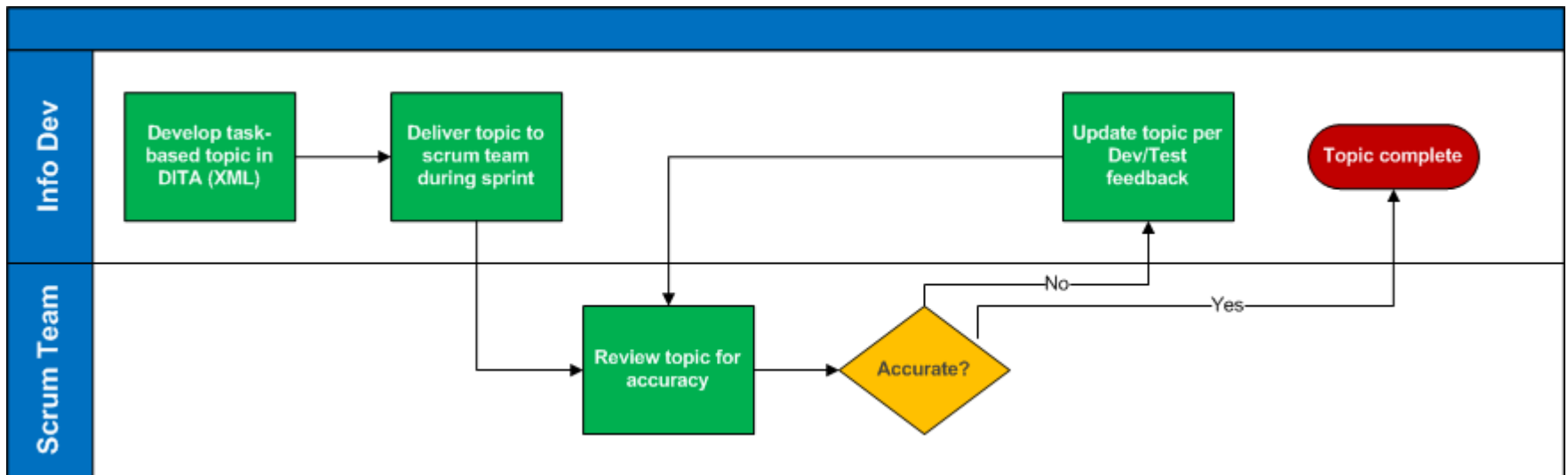
Validation



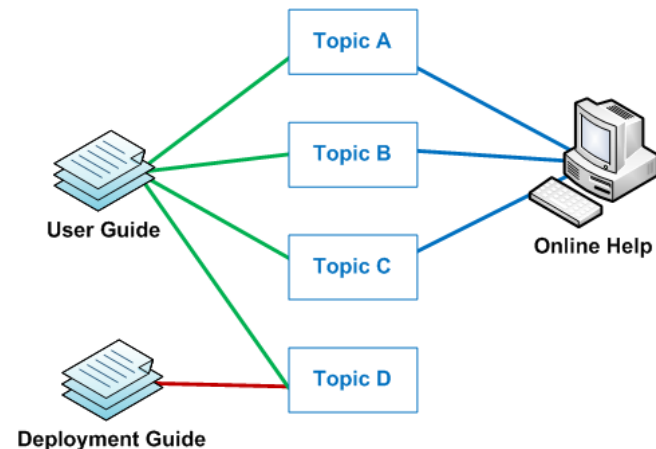
Key Validation Considerations:

- **Review Cycle**
 - One full end-to-end review of Guide content during hardening, and then approval

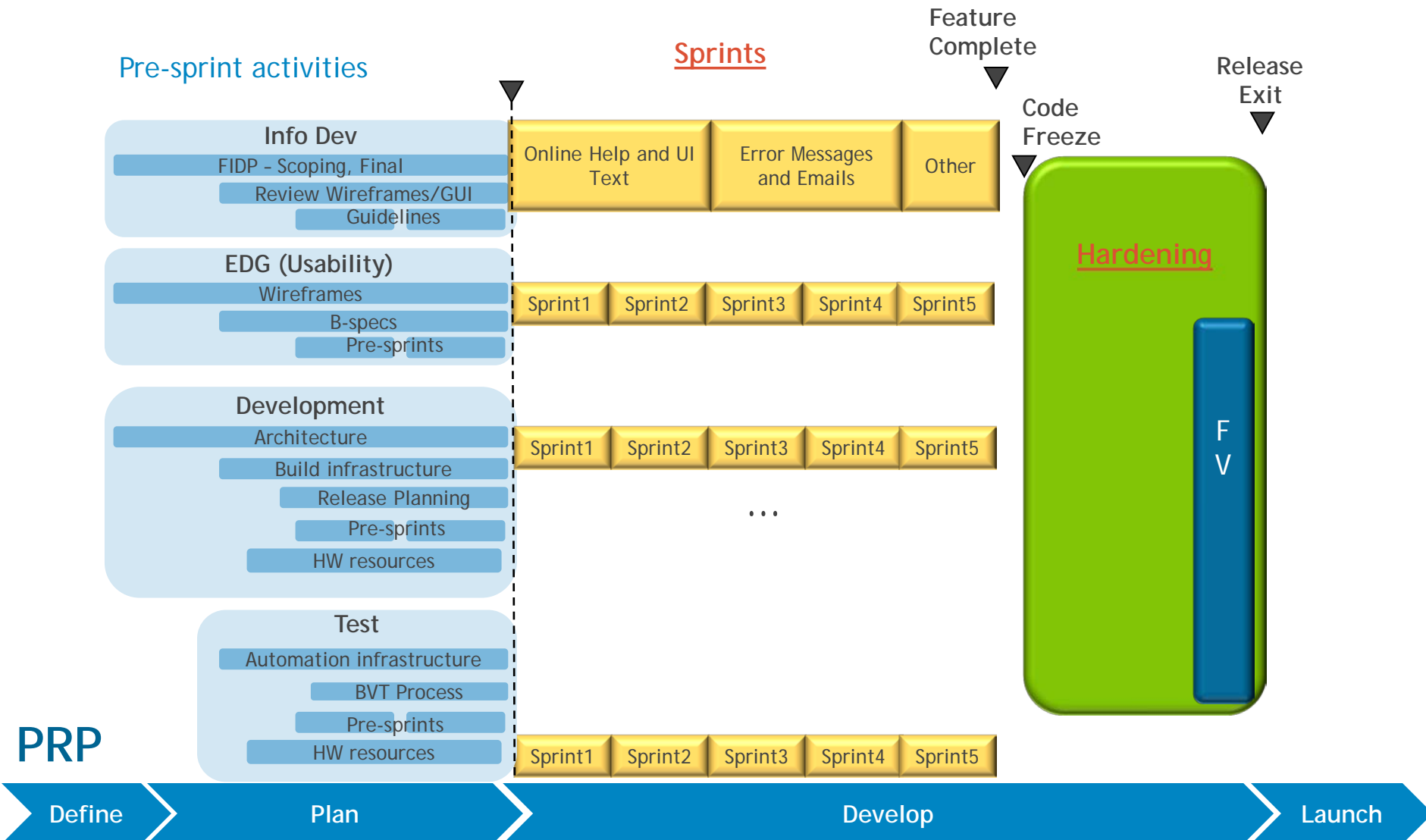
Future Authoring Process



- Follows standard Agile documentation process
- Write/review **once**, use anywhere!
- Piloted on one project first
- Rolled out Q3 of last year



Future Model - Agile for Info Dev



Future Info Dev Staffing Ratios

Team depends on project size and complexity:



- New project (1.0)
One writer per 1 - 3 scrum teams
- Feature release on existing project
One writer per 5 - 7 scrum teams
- Point/minor release on existing project
One writer per 1 - 2 products

*Staff to meet needs
NOT
budgets!!*

Scrum Responsibilities for Artists

- ✓ Collaborate with scrum teams and ID writers to determine art requirements
- ✓ Create illustrations, topologies, flowcharts, etc., to be used in product documentation
- ✓ Create how-to videos or other graphic deliverables

Scrum Responsibilities for Writers

- ✓ Represent Info Dev in Scrum, Scrum-of-Scrum, and Extended Engineering Team meetings, as needed
- ✓ Collaborate with scrum teams to understand details of information deliverables
- ✓ Create information deliverables including Online Help, User's Guide, Deployment Guide, Read Me, System-Generated Emails
- ✓ Edit error messages per Error Message Guidelines
- ✓ Provide artist with information about art-related deliverables, including timeframes, technical specs, and overall product concept

Scrum Responsibilities for Lead Writers

- ✓ Same as writer, plus...
- ✓ Drive day-to-day documentation activities
- ✓ Act as primary POC for project Core Team, including attending Core Team meetings
- ✓ Create and update Functional Information Development Plan (FIDP) as project evolves
- ✓ Ensure Info Dev deliverables are provided to Scrum teams according to agreed-upon deadlines

Scrum Responsibilities for Strategist/Manager

Strategist

- ✓ Collaborate with Core Team to understand overall project scope and high-level deliverables, use as inputs into overall Info Dev strategy
- ✓ Scope Info Dev deliverables, and hand off to writing team

Manager

- ✓ Collaborate with Strategist and Core Team to understand overall project size and concept
- ✓ Assign Info Dev staff according to pre-defined ratios

Info Dev Roles and Responsibilities

Role	Responsibility	During Sprints...
Artist	<ul style="list-style-type: none">• Create artwork	<ul style="list-style-type: none">• Interact with scrum teams and writers to understand art needs• Create illustrations, topologies, etc., per Dev/Writer feedback
Writer	<ul style="list-style-type: none">• Work with scrum teams to determine use cases that must be documented• Develop task-based and system-related documentation	<ul style="list-style-type: none">• Attend scrum, scrum of scrum, extended engineering meetings as needed• Create Online Help and system-generated emails• Edit and help create error messages and UI text
Lead Writer	<ul style="list-style-type: none">• Same as writer responsibilities, plus...• Act as POC for core team• Drive day-to-day doc strategy	<ul style="list-style-type: none">• Attend core team meetings to understand high-level project deliverables and timelines• Develop and update FPP per project evolution
Strategist	<ul style="list-style-type: none">• Scope overall deliverables	<ul style="list-style-type: none">• Attend requirements/planning meetings to understand project size, concept, etc.• Determine initial project scope and hand off to writers
Mgr.	<ul style="list-style-type: none">• Assign team to project	<ul style="list-style-type: none">• Determine staff requirements based on project size• Assign writer, artist, lead writer, strategist, as needed

Conclusion and Q&A

Three key take-aways to be successful with Agile...

#1 - Use a DITA authoring process/toolset that enables iterative content creation and delivery.

#2 - Update traditional deliverables and methods to use the Agile methodology for all project work.

#3 - Use appropriate staffing ratios based on project size/complexity to ensure proper coverage of upcoming projects.

THANK YOU FOR YOUR ATTENTION!!