

Systems Analysis and Design in an Age of Options

G. Spurrier and H. Topi
Chapter 6—Project Planning and
Creating the Product Backlog
© Prospect Press

Chapter 6: Project Planning and Creating the Product Backlog

...

Systems Analysis and Design
in an Age of Options

Learning Objectives

- Compare traditional vs. agile project management, including:
 - Key techniques
 - Appropriateness for agile and hybrid approaches
- Skill: Determine the project management approach and techniques to use in a given project
- Understand project scoping using epics, user stories, acceptance criteria, and prioritization using the MoSCoW model
- Skill: Create a product backlog and agile release plan consisting of prioritized epics and user stories

Introduction

- **How to plan and integrate systems project activities:**
 - **Hybrid, pre-construction planning activities:**
 - Supporting BRUF/BDUF for Business Analysis and Functional & Technical Design
 - Using traditional, plan-driven project planning tools and techniques
 - **Iterative (sprint-based) software construction:**
 - Creation of product backlog and initial sprint planning
 - Using non-traditional, agile planning tools and techniques
- **Focus here on software construction projects:** In contrast, COTS software configuration projects tend to be highly plan-driven because of repeatable steps

Project Planning for Hybrid Software Construction Projects

Need to combine

- **Traditional project techniques:**

For planning

- Visioning
- BRUF/BDUF
- Business case/project charter
- Etc.

- **Non-traditional project techniques:**

For planning

- Product backlog
- Construction iterations/sprints

Systems Development Process Framework: Hybrid Approach

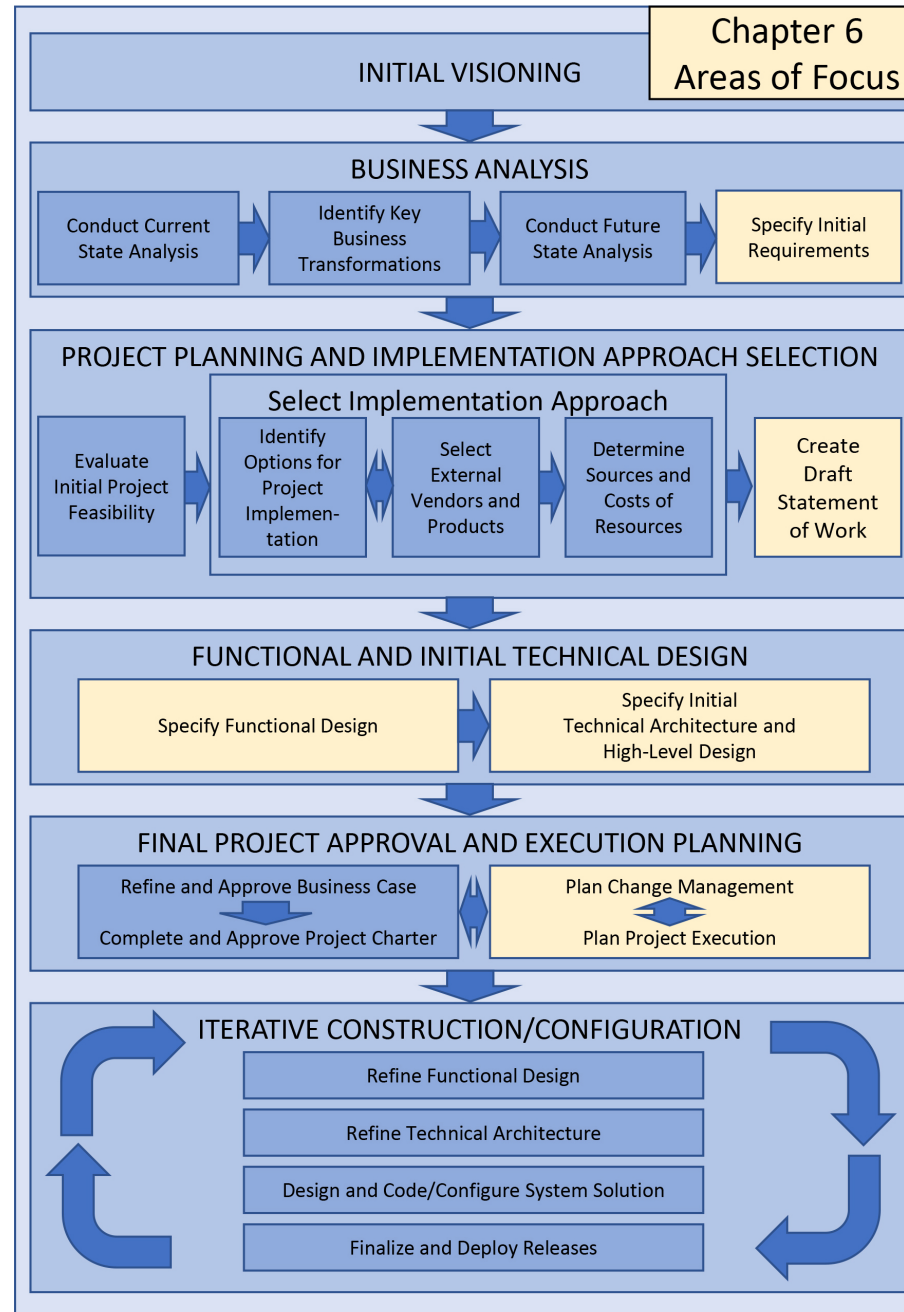


Figure 6-1
Systems Development Process Framework using the hybrid approach, highlighting BRUF/BDUF, followed by product backlog and release planning ● 6-5

Gantt Charts

Traditional Project Planning Tool

- **Mathematically-driven**
- **Good for highly predictable projects:**
 - E.g. building multiple houses using same blueprint
 - Minimizing replication risk
- **Bad for highly unpredictable projects:**
 - Constructing new software features
 - Facing high design risk

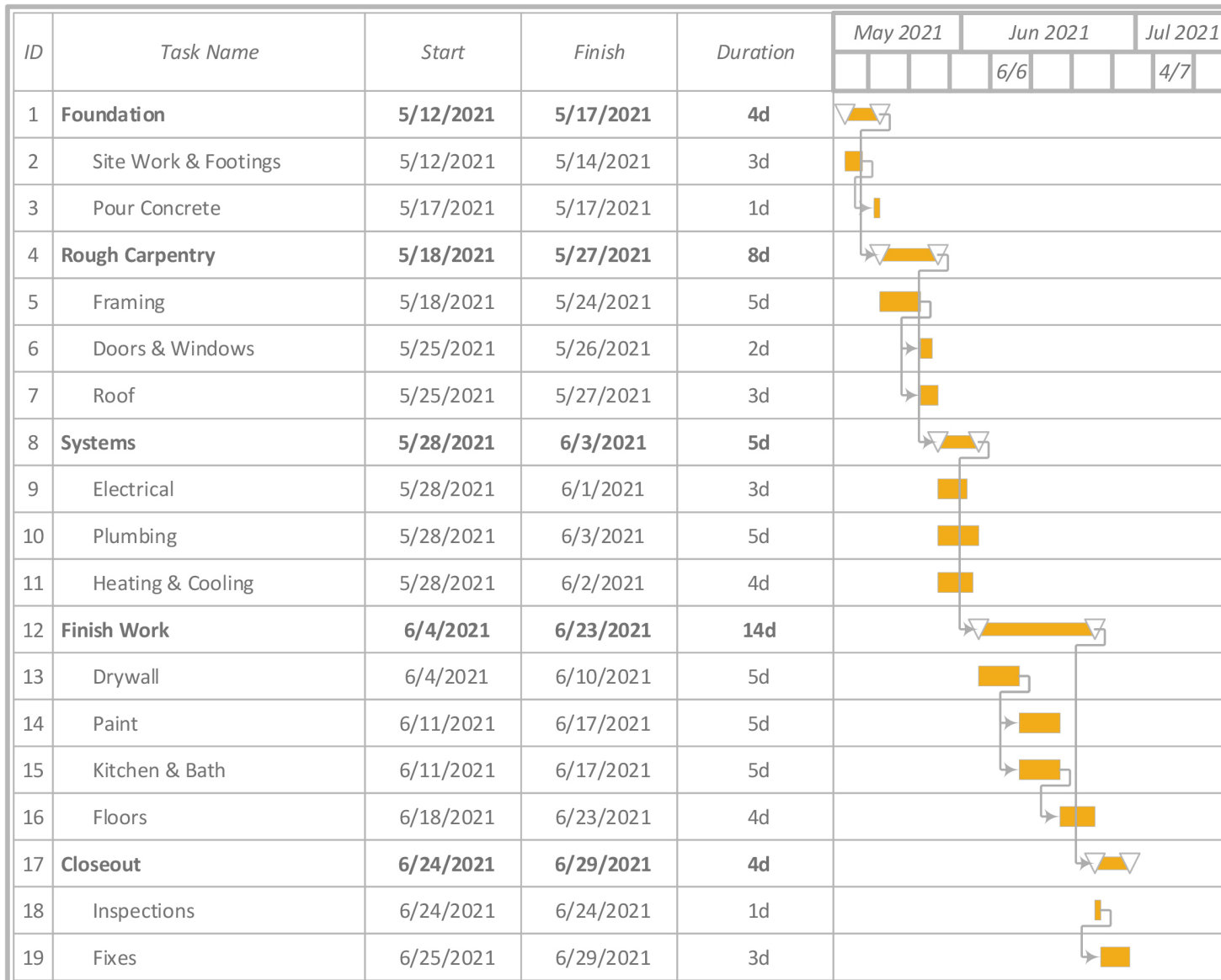


Figure 6-2 Example Gantt chart for managing the construction of a house

Unpredictability of Systems Project Planning

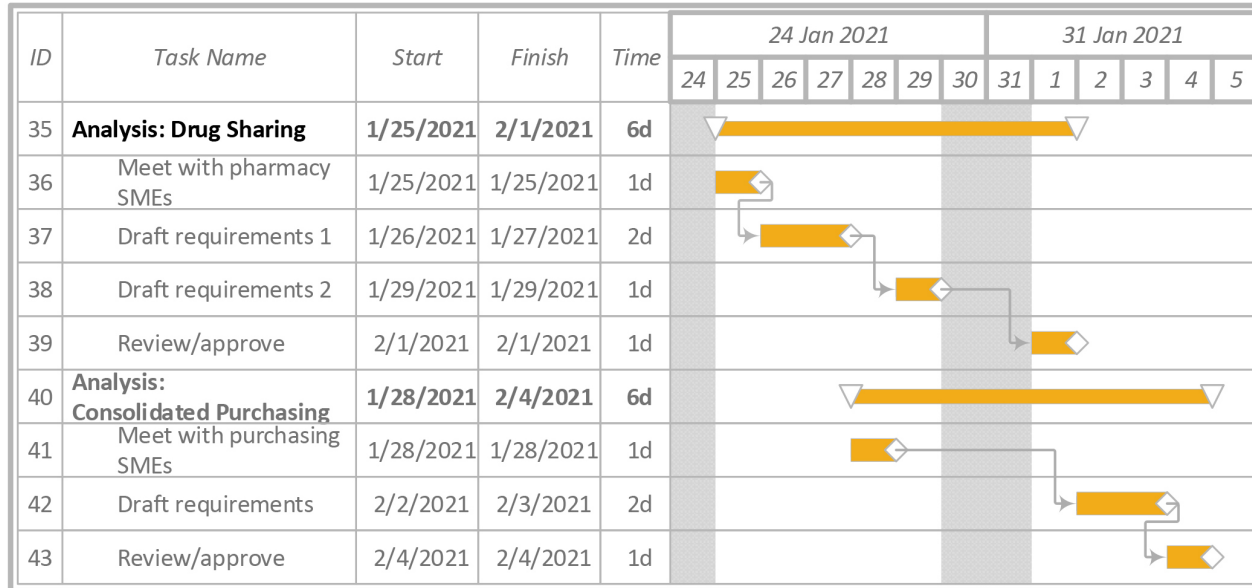


Figure 6-3 Classic Gantt chart for a systems project showing a small subset of tasks and dependencies

SA&D requirements activities highly unpredictable

- Time/effort needed for
 - Business analysis and functional/technical designs?
 - How large and complex are the requirements?
 - How many meetings?
 - Impacts of uncertain user availability, interruptions, rescheduling, etc.
- Difficult to plan and the plan frequently changes



Figure 6-4 Planning and replanning project tasks, rather than creating working software (iStock.com/Kwangmoozaa)

Planning and Re-Planning Project Tasks

Traditional project planning can collapse because of high uncertainty

- **Requirements:** Per prior slide
- **Construction:** Need for flexible, iterative software construction

Key reasons why agile approach advocates argue:

- **Planning:** Is a waste of time
- **Instead focus:** On creating working software

But then how to plan pre-construction activities...?

Predicting the Unpredictable with Gantt-lite Charts

Meaningful but not precise project plans—way to merge traditional and agile project planning techniques

- **Similar to traditional Gantt charts:**
 - Project tasks over time
 - Team member assignments
 - Percentage task completion
- **Differ from traditional Gantt charts:**
 - Not mathematically driven
 - Driven by judgment, experience, and negotiation
 - Task precedences fuzzy but meaningful

	% Complete	Resources	11-Jan-21	18-Jan-21	25-Jan-21	1-Feb-21	8-Feb-21	15-Feb-21	22-Feb-21	1-Mar-21	8-Mar-21	15-Mar-21	22-Mar-21	29-Mar-21	5-Apr-21	12-Apr-21	19-Apr-21	26-Apr-21	3-May-21	10-May-21	17-May-21	24-May-21	31-May-21	7-Jun-21	14-Jun-21	21-Jun-21	28-Jun-21	5-Jul-21	12-Jul-21	19-Jul-21	26-Jul-21	2-Aug-21	9-Aug-21	16-Aug-21	23-Aug-21	30-Aug-21	6-Sep-21	13-Sep-21	20-Sep-21	27-Sep-21	4-Oct-21												
PHASE 1: Vision, Business Analysis, Project Planning																																																					
Project Initiation, ID Stakeholders, Initial Roles, Time Commitments	100	COO, CIO, BA, ProdOwner	✓																																																		
Visioning: JAD Meetings	100	ProdOwner, BA, SMEs		✓																																																	
Visioning: Draft Vision Document	100	ProdOwner, BA			✓																																																
Visioning: Review and Approve Project Vision Document	100	ProdOwner, SMEs, COO, CIO				✓																																															
Business Analysis (Area 1)	100	BA, ProdOwner											✓	■																																							
Business Analysis (Area N)	100	BA, ProdOwner					✓																																														
ID Stories and Acceptance Criteria	100	BA, ProdOwner																																																			
Review and Refine Product Backlog, ID High Risk Scope	100	BA, ScrumMaster, Devs											✓																																								
Estimate Product Backlog Costs	20	BA, ScrumMaster, Devs																																																			
Estimate Business Benefits	25	ProdOwner, BA, COO																																																			
Draft SOW, Review and Approve	10	BA, CIO, COO, ProdOwner																																																			
PHASE 2: Functional Design and Architecture																																																					
Draft Use Case Narratives	60	BA, ProdOwner																																																			
UI Technical Design	30	BA, ProdOwner																																																			
Review and Approve Product Backlog	10	BA, ProdOwner, COO																																																			
Re-estimate Product Backlog Costs	20	BA, ScrumMaster, Devs																																																			
Re-estimate Business Benefits	25	ProdOwner, BA, COO																																																			
Draft and Approve Business Case	10	BA, ProdOwner																																																			
Draft and Approve Project Charter	0	BA, CIO, COO ProdOwner																																																			
Plan Execution of Change Management and Backlog Items																																																					
PHASE 3: Sprint Construction																																																					
Sprint 1	0	IT Team & ProdOwner																																																			
Sprint 2	0	IT Team & ProdOwner																																																			
Sprint 3	0	IT Team & ProdOwner																																																			
Sprint 4	0	IT Team & ProdOwner																																																			
Sprint 5	0	IT Team & ProdOwner																																																			
Sprint 6	0	IT Team & ProdOwner																																																			
PHASE 4: Change Management, UAT and Deployment																																																					
Change Management Planning	0	COO, ProdOwner, BA																																																			
Policies & Procedures Updates	0	ProdOwner, COO																																																			
Execute Training	0	ProdOwner, SMEs																																																			
Implement Business Data Updates	0	ProdOwner, SMEs																																																			
Plan UAT	0	ProdOwner																																																			
Execute UAT	0	ProdOwner, SMEs																																																			
Deployment	0	BA, ProdOwner, SMEs, DevOps, Devs																																																			
Begin Enhanced Support	0	BA, ProdOwner, SMEs, DevOps, Devs																																																			

Figure 6-5 Gantt-lite chart

Defining Product Scope

- Plan-driven Fixed Scope
VS.
- Hybrid Semi-Flexible Scope
VS.
- Agile Flexible Scope

Software Approach Dimension	Plan-Driven Approach (Traditional SDLC or Waterfall)	Hybrid Approach	Agile Approach
REQUIREMENTS			
Up-front Reqs Docs	Comprehensive BRUF and BDUF <ul style="list-style-type: none"> • Fully detailed features and designs 	Intelligent Modeling BRUF and BDUF <ul style="list-style-type: none"> • Identify user stories • “Intelligent modeling” detailing designs where it adds value 	Develop Product Backlog <ul style="list-style-type: none"> • Identify user stories • Defer detailed designs until construction
Up-front Scope	Fixed Scope <ul style="list-style-type: none"> • No scope changes once project approved 	Semi-flexible, “guardrails” scope <ul style="list-style-type: none"> • Minimum Viable Product (MVP) WITH • Maximum scope (“Won’t Haves”) 	Flexible Scope: <ul style="list-style-type: none"> • Prioritized stories • Optional: May use a minimum scope (MVP)
On-going Reqs	None <ul style="list-style-type: none"> • Single, long-phase of construction precludes on-going requirements 	Customer feedback via Sprint Reviews: <ul style="list-style-type: none"> • Course corrections and “tweaks” • Stay within guardrails scope 	Emergent Requirements <ul style="list-style-type: none"> • Detailed requirements here • “Just barely good enough” requirements • Customer feedback via Sprint Reviews
CONSTRUCTION			
	Non-Iterative: <ul style="list-style-type: none"> • Single-phase • Long-duration • Testing on delivery • Customer feedback on delivery 	Iterations/Sprints	Iterations/Sprints

Table 6-1

Using Activity Diagrams to Identify User Stories

Approaches include:

- **Using Activity Diagrams:** to ID user stories within a business process epic
 - User/system interactions
 - System automated functions
- **Identifying collections of related but otherwise independent user stories:**
 - Individual reports in a data warehouse

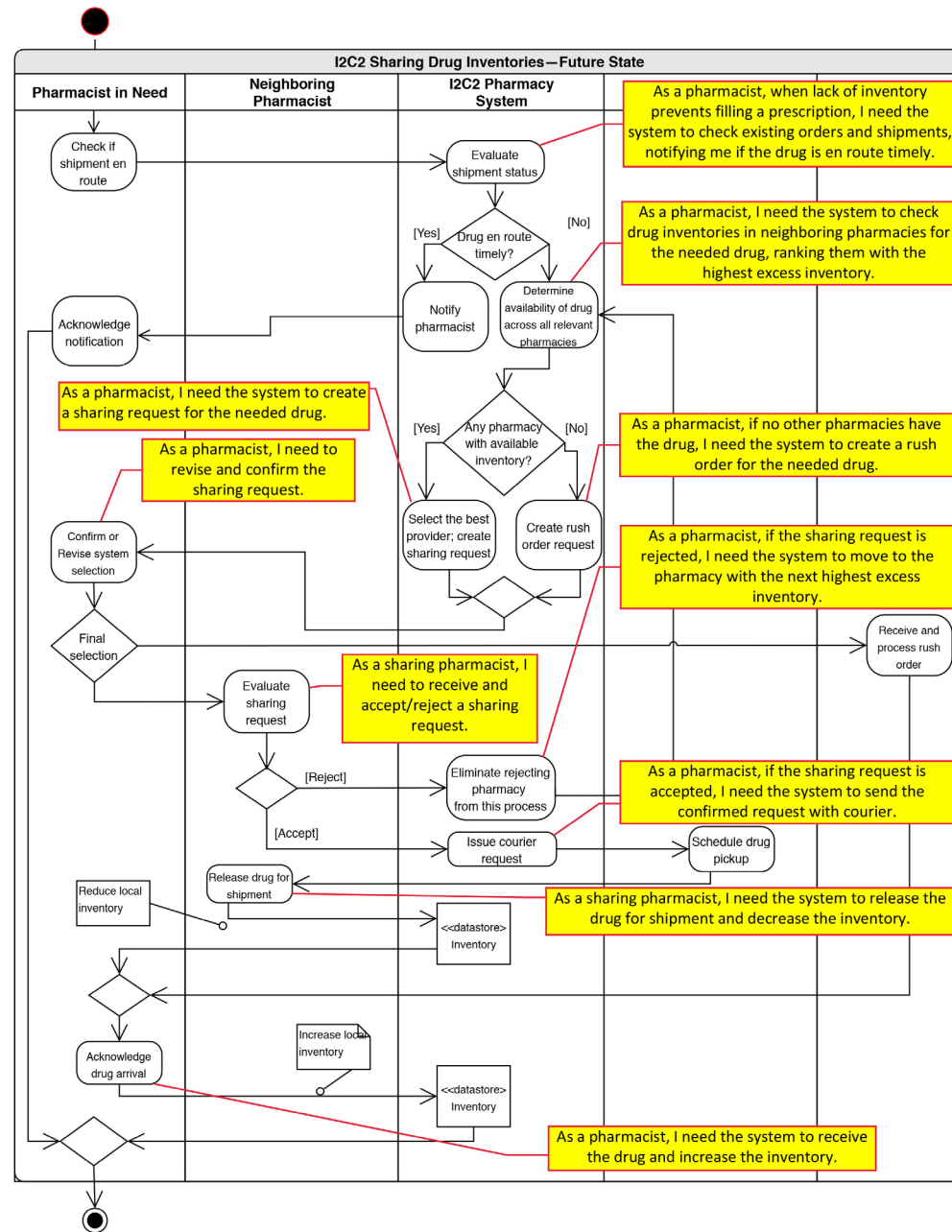


Figure 6-6 Identifying user stories from an activity diagram

Agile Release Planning: Assumptions and Approach

- **Product Backlog exists:**
 - User stories
 - Prioritized/Ranked 1 to N
 - User stories are independent
 - User stories are small (each can fit into a single sprint)
- **Sprints:**
 - Fixed length (1 to 4 weeks each)
 - Capacity = Length X Developers
- **Approach:**
 - Pick the highest priority story for Sprint X
 - Repeat until Sprint X is full
 - Only plan one sprint at a time (maintain flexibility for changing stories and priorities)

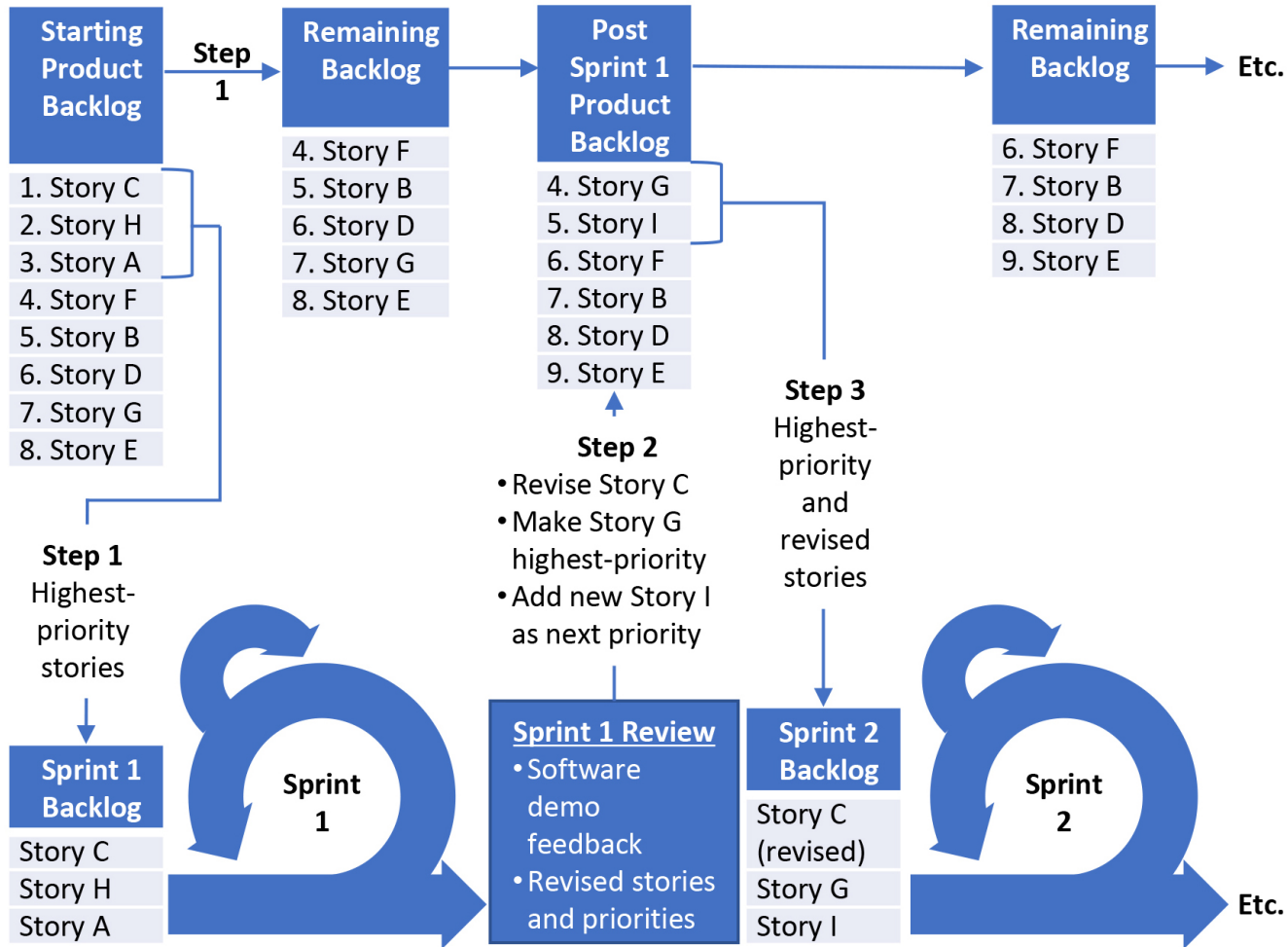
Agile Approach to Release Planning

Step 1: Assign the highest-priority product backlog stories to the Sprint 1 Backlog

Step 2: After Sprint 1, review the process updates remaining in the product backlog

Step 3: Assign the (revised) highest-priority product backlog stories to the Sprint 2 Backlog

Steps 4 and beyond: Pattern repeats for all remaining sprints



Agile Sprint-by-Sprint Release Planning

Used in highly agile projects:

- Small, simple scope
- Flexible scope
- Independent user stories

Figure 6-7 Agile sprint-by-sprint planning

Principles of Hybrid Release Planning

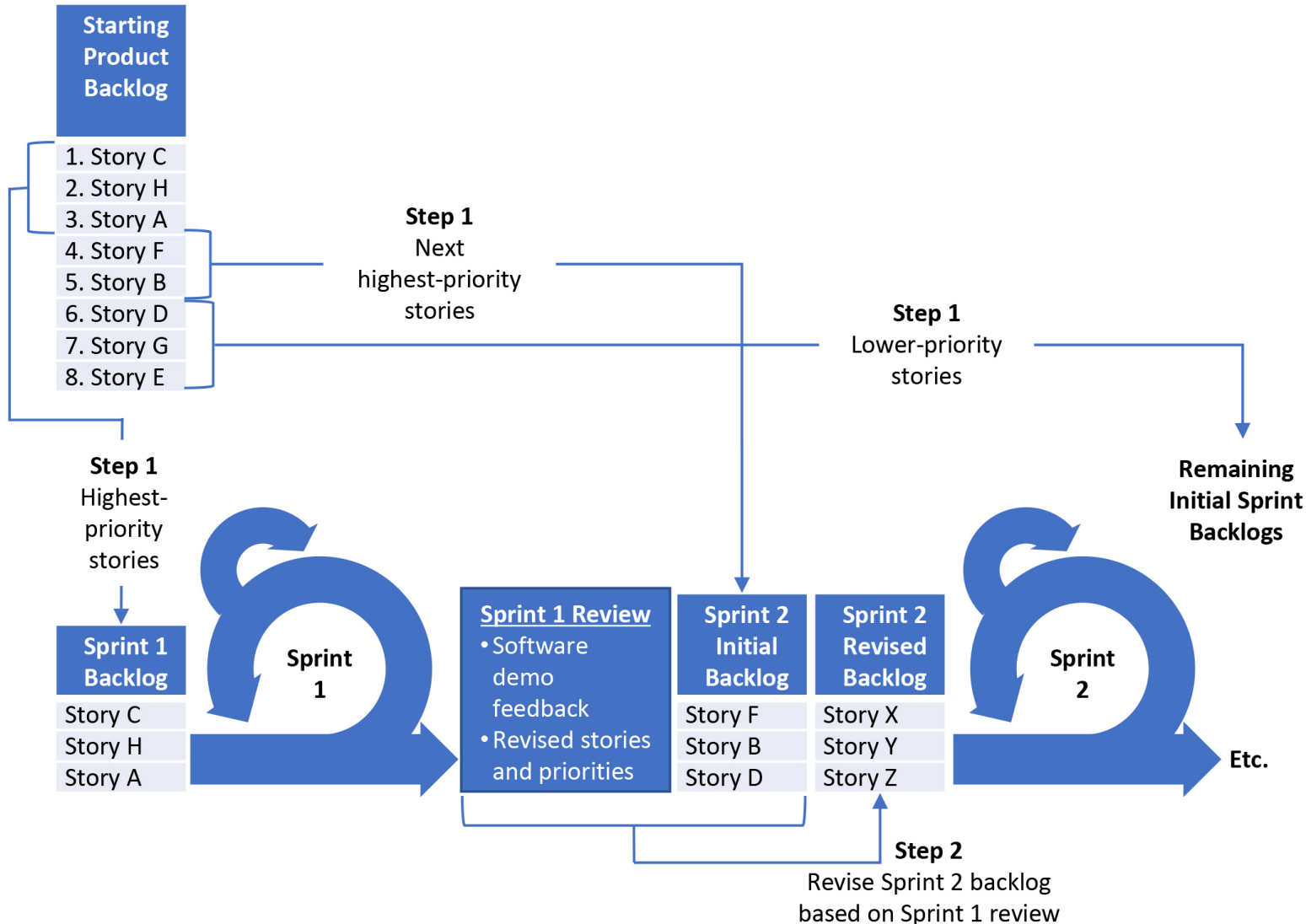
- **Prioritizing large number of stories:**
 - If dozens of stories, how much more important is, say, priority 37 vs. 38?
 - More likely to use MoSCoW model
- **Thinking in terms of epics (rather than individual user stories):**
 - Often it is MVP of entire epic (rather than one user story) that adds business value
 - So doesn't make sense to do one story from Epic A and another story from Epic B and yet another story from Epic C
 - Instead, do all the Must Haves from single most important epic, then Must Haves from next most important epic, etc.
 - Once the Must Haves are done, cycle through on the Should Haves
 - Repeat until scope complete or project time/budget exhausted

Hybrid Approach to Release Planning

Step 1: Assign all stories to sprints based on priorities, creating initial sprint backlogs

Step 2: After Sprint 1, revise the Sprint 2 initial backlog based on the Sprint 1 review

Steps 3 and beyond: Pattern repeats for all remaining sprints.

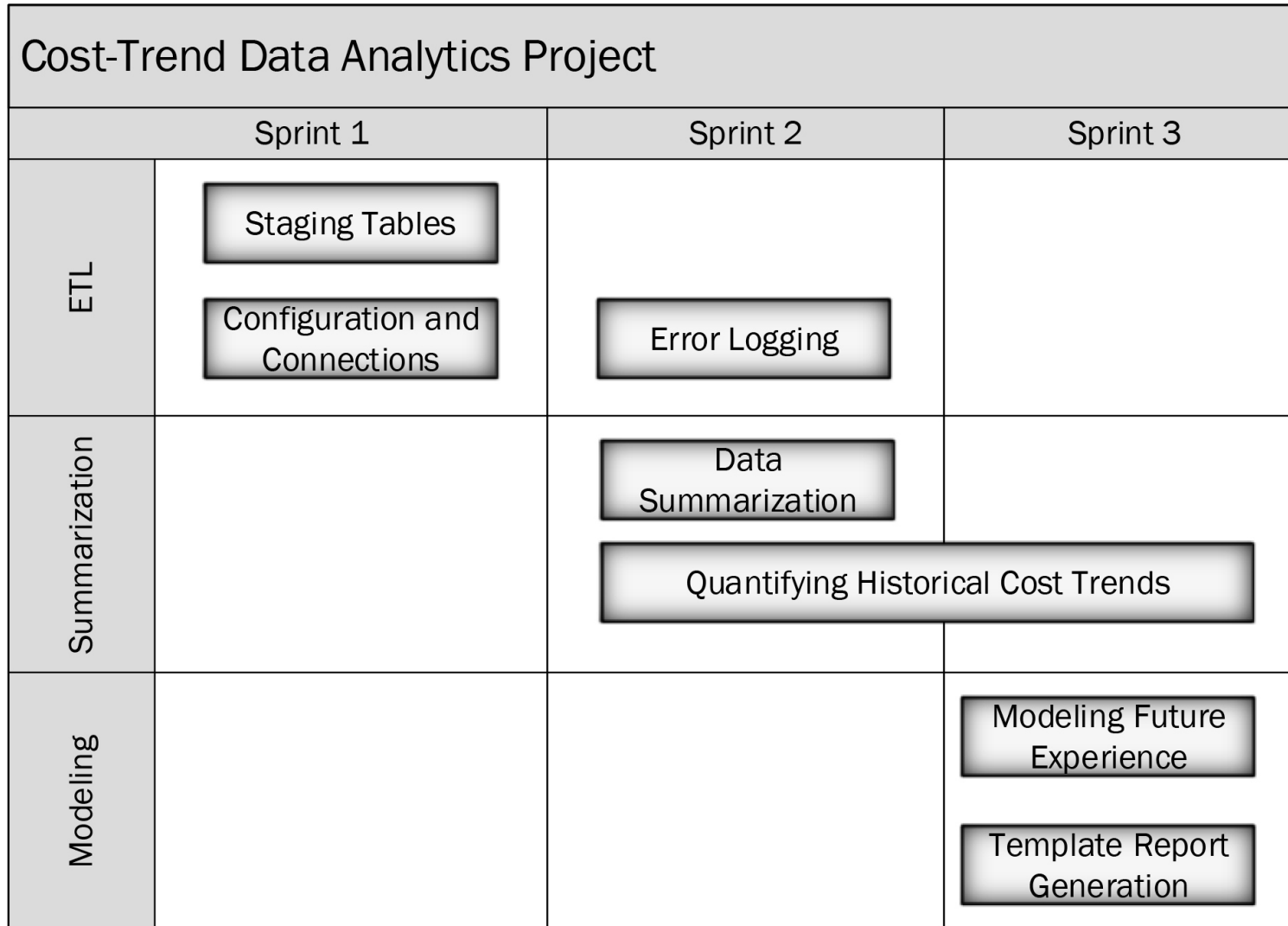


Hybrid Release Planning

Used in hybrid projects:

- Large, complex scope
- Semi-flexible scope with Minimum Viable Product (MVP)
- Interdependent user stories

Figure 6-8 Hybrid release planning



Agile Release Plan (AKA “Flight Plan”)

High-level tool to organize and visualize release plan:

- Gantt chart-like
- Time moving left to right by sprint
- Epics on vertical axis
- Shows logical segments of user stories within an epic

Figure 6-9 Agile release plan (also called a flight plan) diagram

Allocating User Stories to Sprints in Hybrid Release Planning

Leverages product backlog:

- User stories
- Prioritized (e.g., MoSCoW model)
- Estimated (e.g., ideal days or hours)
- Split to fit into a single sprint
- Dependencies identified

Epic	Story ID	Story Description	MoSCoW Priority	Dependencies	Ideal Days Estimate	Sprint
Consolidated Purchasing	CONPUR1	System auto-schedules drug purchasing analysis.	SH		2	3
	CONPUR2	System analyzes past and future drug utilization for each pharmacy and drug.	MH		5	1
	CONPUR3	System identifies volume discounts at current and increased ordering levels.	MH	CONPUR2	5	1
	CONPUR4	System creates report of recommended volume discounts to pharmacy lead.	SH	CONPUR3	2	1
	CONPUR5	Pharmacy lead approves or updates system ordering recommendations.	MH	CONPUR4	2	1
	CONPUR6	Systems allocates and sends updated order quantities to individual pharmacists.	NH	CONPUR5	2	3
	CONPUR7	Individual pharmacist receives order quantities and responds timely to pharmacy leader.	NH	CONPUR6	2	3
	CONPUR8	Using responses, pharmacy lead finalizes drug orders.	NH	CONPUR7	2	3
	CONPUR9	System finalizes order allocations to pharmacies.	NH	CONPUR8	1	3
	CONPUR10	System sends drug orders to each drug vendor.	SH		2	3
	CONPUR11	On order receipt, system checks drugs received vs. ordered and updates inventories.	NH	CONPUR10	3	4
Consolidate Purchasing Epic: Total Ideal Days Estimate					28	
Drug Sharing	SHARE1	When lack of inventory prevents filling prescription, system checks existing orders and shipments, notifies Pharmacist if en route timely.	MH		3	1
	SHARE2A	System reads cross-pharmacy DB for drug inventories of needed drug.	MH	SHARE1	3	1
	SHARE2B	System IDs pharmacies with excess inventory(s), ranking by highest excess inventory.	MH	SHARE2A	3	2
	SHARE3	System creates rush order request.	SH	SHARE2B	2	4
	SHARE4	System IDs excesses and creates sharing request.	MH	SHARE1	2	2
	SHARE5	Pharmacist revises and confirms sharing request.	MH	SHARE4	1	2
	SHARE6	Sharing pharmacist receives and accepts/rejects sharing request.	MH	SHARE5	1	2
	SHARE7	If sharing request rejected move to next highest excess inventory pharmacy.	MH	SHARE6	1	2
	SHARE8	If sharing request accepted issue sharing request.	SH	SHARE6	1	4
	SHARE9	Arrange courier and decrease inventory.	SH		2	4
SHARE10	Receive drug, match order, increase inventory.	SH		1	4	
Drug Sharing Epic: Total Ideal Days Estimate					20	
Drug Utilization Analysis	UTILIZ1	System auto-schedules drug utilization analysis.	SH		2	4
	UTILIZ2	System polls each pharmacy DB for drug utilization by month for up to 24 months.	MH		5	2
	UTILIZ3	Generate maintenance drug point historical utilization and future trend.	MH	UTILIZ2	4	2
	UTILIZ4	Create maintenance drug utilization dashboard.	MH	UTILIZ3	2	2
	UTILIZ5A	Generate non-maintenance drug historical utilization and future trend.	MH	UTILIZ2	4	3
	UTILIZ5B	Generate non-maintenance drug utilization seasonal adjustments.	MH	UTILIZ5A	2	3
	UTILIZ6	Create non-maintenance drug utilization dashboard.	MH	UTILIZ5B	2	3
	UTILIZ7	Pharmacy leader revises and approves drug trend factors for consolidated purchasing.	SH	UTILIZ3, UTILIZ5	2	4
	UTILIZ8	Update CONPUR2 analysis to include utilization trend analysis.	SH	UTILIZ7	4	4
Drug Utilization Analysis Epic: Total Idea Days Estimate					27	
Product Backlog: Total Ideal Days Estimate					75	

Figure 6-10 I2C2 Pharmacy enhancement epics, stories, priorities, dependencies, estimates, and initial sprint assignments

Hybrid Release Planning: Preparing to Allocate Stories to Sprints

- Assumes already determined:
 - User stories
 - Prioritized (e.g., MoSCoW model)
 - Estimated (e.g., ideal days or hours)
 - Split to fit into a single sprint
 - Dependencies identified
- Then:
 - Prioritize the epics
 - Story interdependencies: Promote lower priority story(s) required for higher priority story
- Allocate user stories to sprints using steps on next slide

Hybrid Release Planning: Steps to Allocate Stories to Sprints

Step	Step Description	Notes
1	For Sprint X	Start with Sprint 1 until full, then move to Sprint 2, etc.
2	For highest MoSCoW priority with unallocated stories	Start with MH until all allocated, then SH, then NH
3	For highest priority epic with unallocated stories at this story priority	Start with highest priority epic and move to lowest priority
4	Pick unallocated user story at this priority (and “promoted” predecessors(s)) and allocate to sprint	
5	If no more unallocated stories in this epic at this priority, then move to next epic	E.g., once highest priority epic has no more MH priority stories, then move to next highest priority epic
6	If no more epics with unallocated stories at this priority, move to next priority	E.g., move from MH to SH
7	If sprint has no more capacity, then move to next sprint	E.g., when Sprint 1 is at capacity, move to Sprint 2

I2C2 Pharmacy Enhancements Project Agile Release Plan (Flight Plan)				
	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Consolidated Purchasing	<div>Past Utilization Analysis</div> <div>Create Purchasing Recommendations</div>		<div>Purchasing Analysis Scheduler</div> <div>Pharmacist Revisions</div>	
Drug Sharing	<div>Drug Inventory Search and Sharing Requests</div>			<div>Rush Orders</div> <div>Drug Sharing Transfers</div>
Drug Utilization Analysis		<div>Read Inventory Data from Pharmacies and Generate Drug Utilization and Trend Analysis</div>		<div>Utilization Scheduler</div> <div>Use Analysis in Consolidated Purchasing</div>

Agile Release Plan (Flight Plan) for I2C2 Pharmacy Enhancements

- Derived from allocated the users stories to spring using hybrid release planning process
- Created meaningful labels for blocks of user stories within each epic

Figure 6-11 Agile release plan (flight plan) for I2C2 pharmacy enhancements

Chapter Summary

...

Let's Review

- Focused on techniques for:
 - Project planning
 - Creating a project backlog
 - Release planning
- Few systems projects present good fit for traditional project management
 - Highly unpredictable
 - Emphasize design work rather than replicating existing designs
- Hybrid systems projects combine:
 - Relatively linear pre-construction activities
 - Iterative construction using sprints
- Systems project planning techniques:
 - Gantt-lite (rather than Gantt) charts for hybrid pre-construction activities
 - Agile release planning for construction
- Project approaches use different scope definitions:
 - Plan-driven: Fixed Scope
 - Hybrid: Semi-flexible Scope
 - Agile: Flexible Scope
- Release planning approaches:
 - Agile: Allocate user stories only for next sprint
 - Hybrid:
 - Allocate user stories to multiple sprints
 - Utilize both epic and user story priorities and dependencies
 - Acknowledge that release plan will evolve as project executes
 - Use agile release plan (flight plan) to visually express overall plan