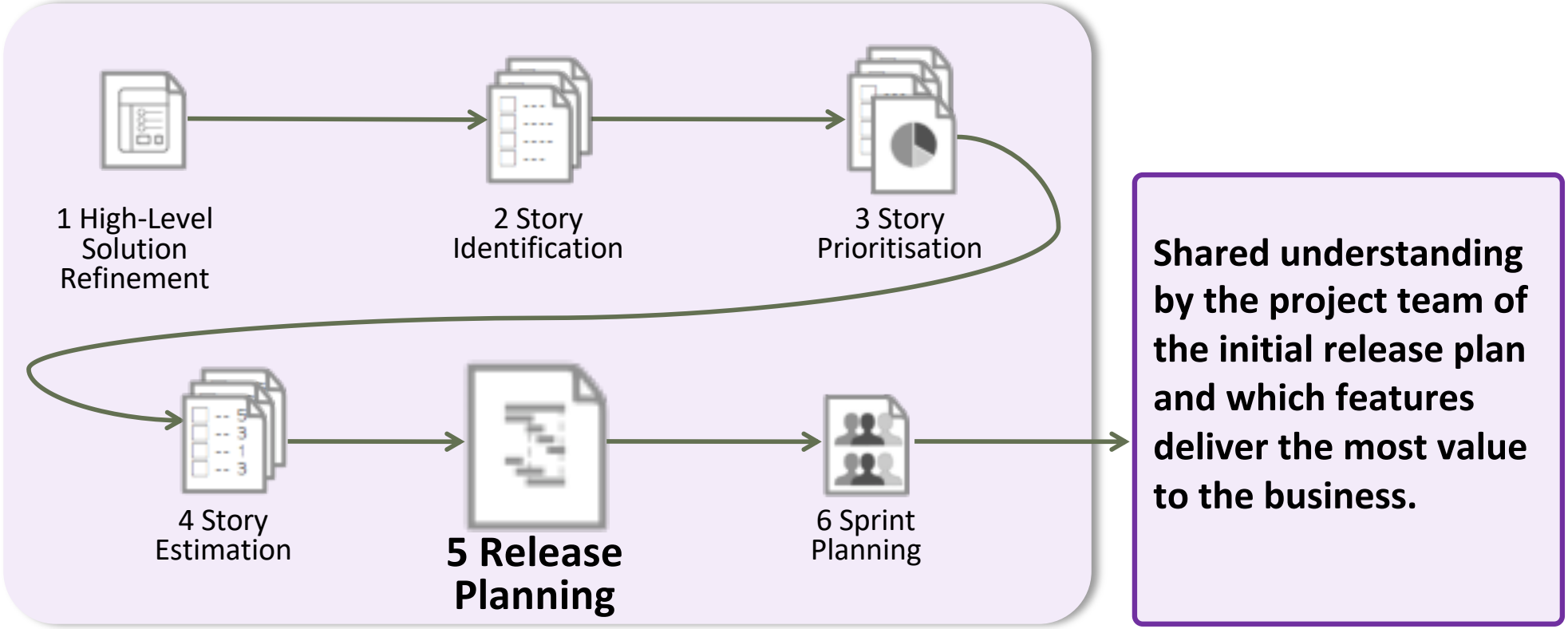
# Release Planning

CSSE3012

# Release Planning

GOAL: Assemble user stories into logical groups for releases and subsequently decomposition of the first release into sprints.



# Release Planning

- Collect user stories into coherent groups of functionality
- Identify the smallest set of stories that deliver immediate business value
  - initial release
  - subsequent releases are smallest increment that delivers additional business value
- Deployment overhead influences the size and frequency of releases

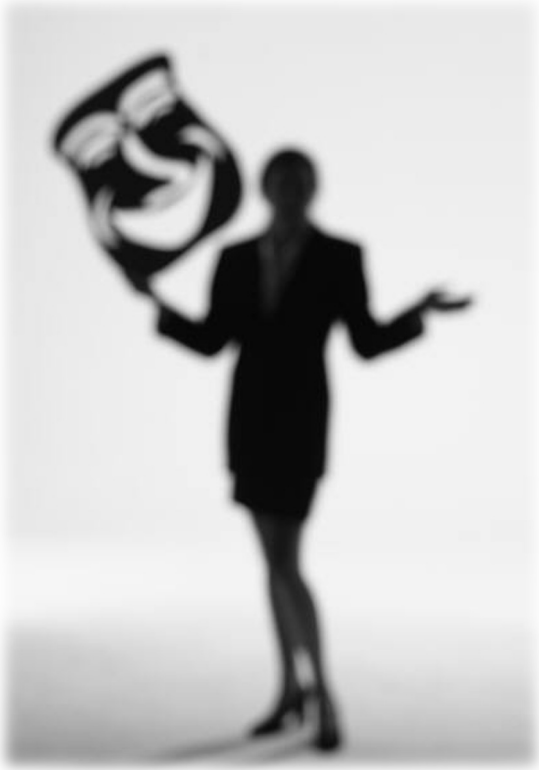
Release Planning

≠

Prioritisation

# Release Planning Roles

- Customer decides on priorities
- Developers provide input



# Release Planning Example

- Imagine user stories for an Internet Banking system
- Let's organise them into coherent releases
- First few releases will be almost exclusively "Must Have" stories
  - a few complementary lower priority stories may be added to satisfy user expectations
- "Won't Have" stories are out of scope

# Internet Banking Example Release Plan

## Internal Release One

- Feasibility Demonstration (30 points)
  - Must Haves from View Accounts, Transfers and BPay
    - without implementing reliability

## Internal Release Two

- Security Base (20 points)
  - most Must Haves from Security

## Release Candidate One (MVP)

- Reliability (18 points)
  - ensure transactions are reliable, plus remaining Security story

# Internet Banking Example Release Plan

## Internal Release Three

- Increased Flexibility (20 points)
  - Should Haves: view transactions in range, schedule for future date, correct mistake, login with account number

## Internal Release Four

- Remember for Me (30 points)
  - Should Haves: maintain external account list, maintain BPay list, bill from existing billers

## Internal Release Five

- Scheduling (18 points)
  - Could Haves, without SMS confirmation and duplicate transaction warnings

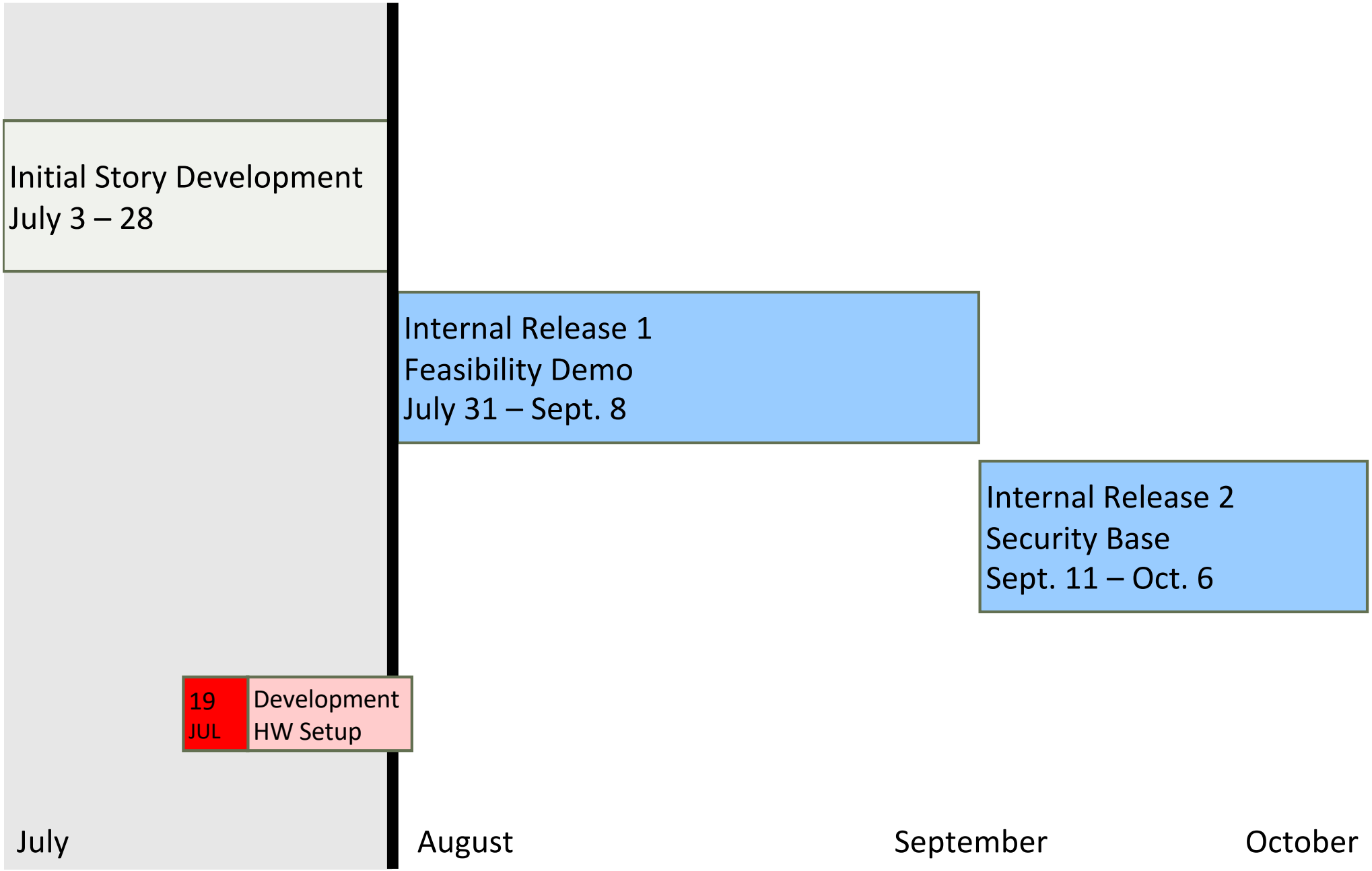
# Internet Banking Example Release Plan

## Release Candidate Two

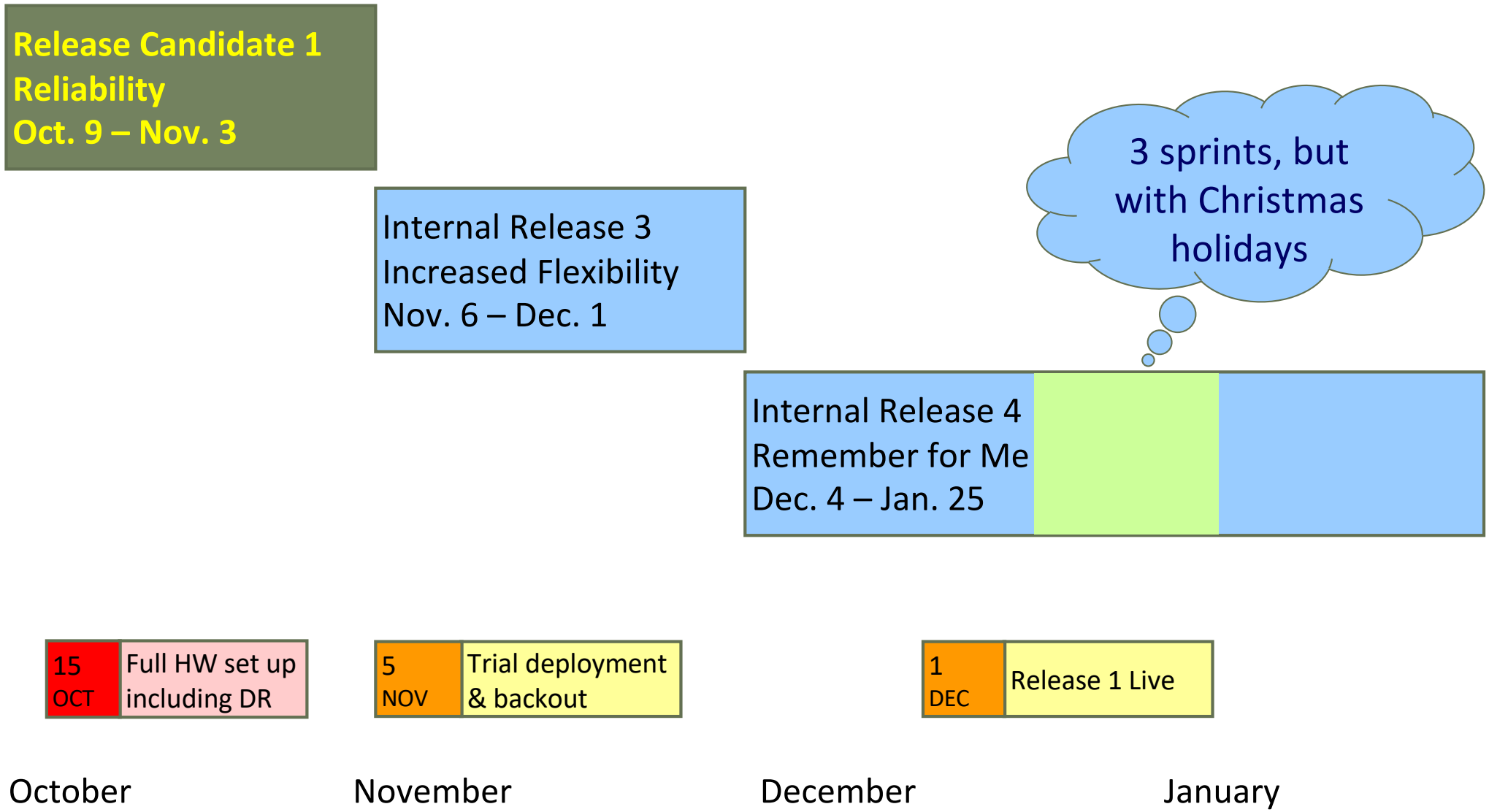
- Extra Security and Assurance (18 points)
  - confirmation of PIN change by SMS and warnings about duplicate transactions

- Story point estimation: 154 points
- Cost estimation calculated a team velocity averaging one story point per day
- Team is following fixed two week sprints

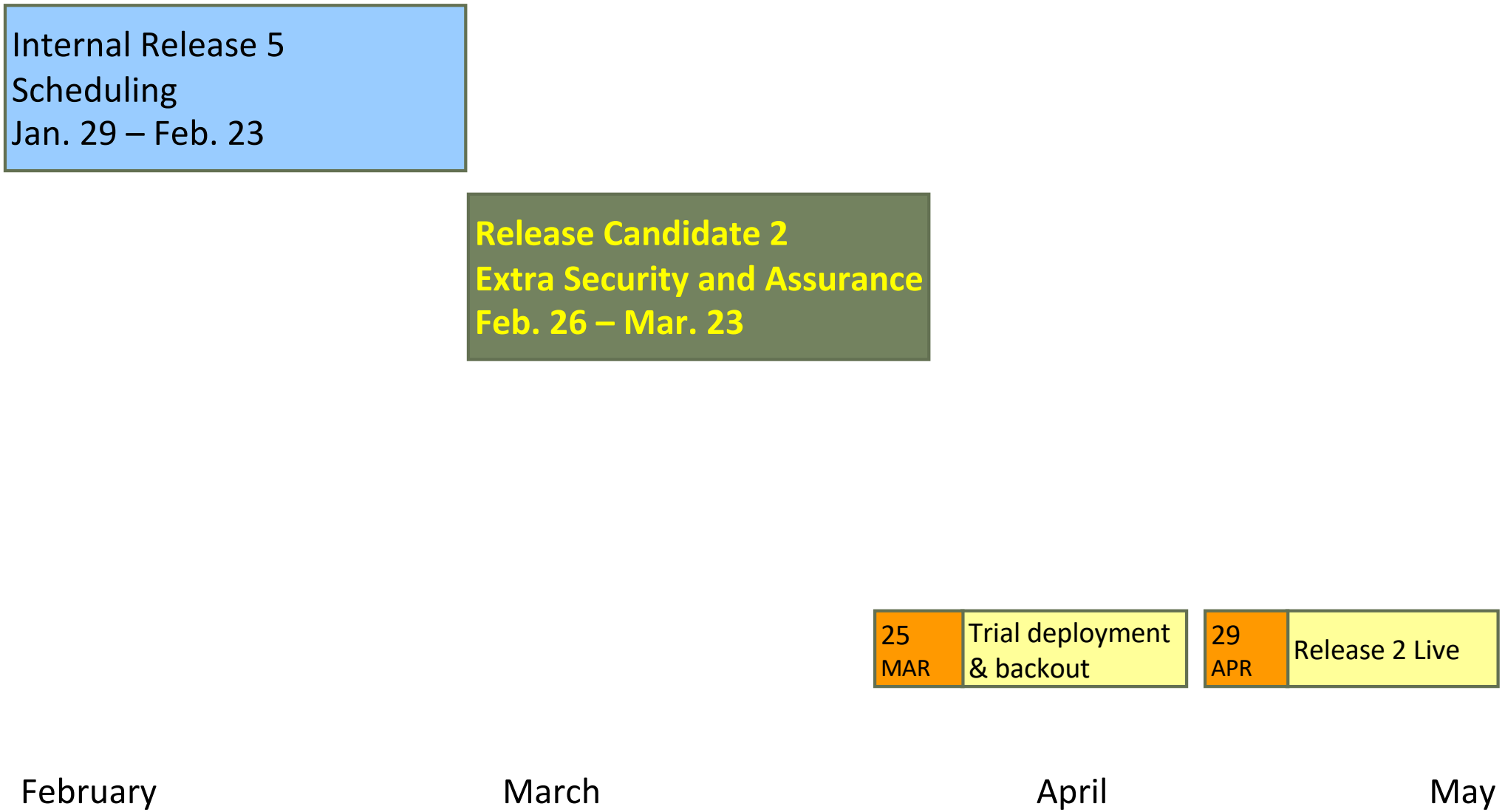
# Internet Banking Example Release Plan



# Internet Banking Example Release Plan



# Internet Banking Example Release Plan



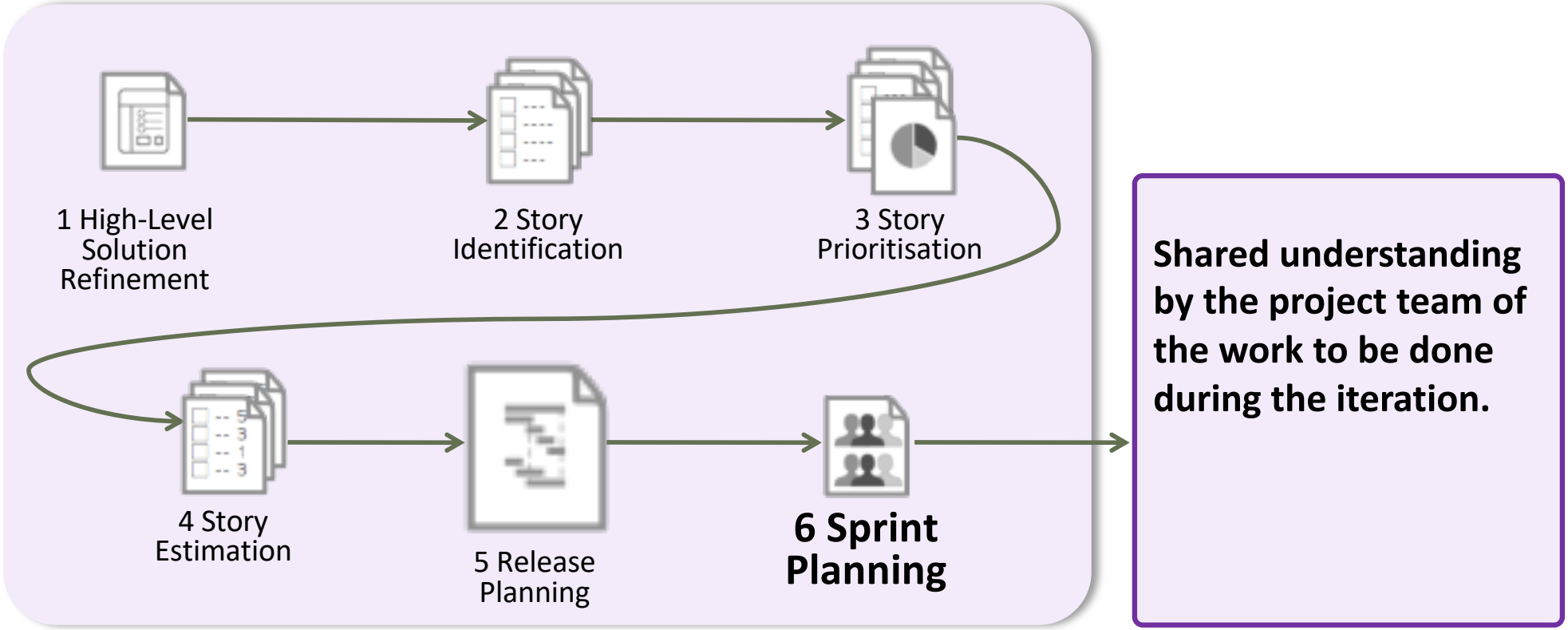
# Release Planning Issues

- Stakeholder key dates
  - What happens if the release plan doesn't correspond to stakeholders' important dates?
    - e.g. Summer Olympics web site by Sept. 2022
- Balancing business value vs. technical risks
- External dependencies
  - risks of delay?
- Resource requirements
  - fully identified?
  - availability?

# Release Plan Template

# Sprint Planning

GOAL: Identify user stories to be completed in an iteration and subsequently decompose them into implementation tasks.



# Sprint Planning

- Planning meeting at the start of a sprint
  - < half-day for a 2 week sprint
- Revisit release plan
  - Does the client have new priorities?
- Review last sprint
  - How many story points completed?
    - update velocity
  - Decide what to do about stories not completed.
    - client decides

# Sprint Planning Activities

- Customer selects stories from backlog
  - cannot exceed story points completed in last sprint
- Customer prioritises stories from highest to lowest
- Team breaks stories into tasks
  - estimating size of each task

# Task Breakdown

- Read out story
- Brainstorm tasks required to implement story
  - keep them short
    - half a day or ***less*** is good
  - write each task on an index card
- All stories have a “Verify story is complete” task
- Review list of tasks
  - does the list seem complete?
- Compare lists of tasks between stories
  - was something forgotten?

# Definition of Done


DoD

- What does “complete” mean?
  - code finished?
  - unit tested?
  - integration tested?
  - acceptance tested?
  - deployed?

# Estimate Tasks

- Back to planning poker …
- Your numbers are now hours
  - 1, 2, 4, 8, 16, 32
- Any task over 4 hours should probably be split
- Review story estimates
  - do the tasks align with the story points?
- Renegotiate stories who's estimate has grown
  - split stories
  - drop stories
  - move to later sprints

# During a Sprint

- Highest priority story is the one under development
- Developers work on 1 task at a time, until done
- Stories are completed throughout the sprint
  - *not* all at the end
- "Next task" is any task from the unfinished story with the highest priority
  - taking into account dependencies between tasks
- When you complete a task mark the card complete
- If you discover a new task create a card for it
  - highlight the card
    - review these in sprint retrospective

# Progress Checkpoint

- Review progress halfway through the sprint
  - are you going to finish all the stories?
- If not, quickly redo sprint plan
  - which stories will likely be dropped?
    - confirm priorities with product owner
  - review task estimates
    - was there consistent under estimation?
- Key principles: Communication & Courage
  - keep the customer informed and on-side

# Sprint Plan Template

# Sprint Retrospective

- Review team's process
  - how to improve
  - not who to blame
- What went well?
  - keep doing this
- What needs improvement?
  - pain points
- Next steps?
  - pick *one* issue to try and fix in next sprint

# Reading

- Sommerville, Chapter 23
- Larman, Chapter 40
- https://www.atlassian.com/team-playbook/plays/retrospective
- https://www.atlassian.com/blog/jira-software/5-fun-sprint-retrospective-ideas-templates

# Next Steps

- Guest Lecture (March 31)
  - Creating Imaginary Things
    - Damien Fitzpatrick
      - Senior Software Development Manager at Amazon
- Tutorial
  - Release and Sprint Planning
- Next Week
  - Use Case Modelling
- Assessment
  - User Stories – April 4, 4:00pm
  - Videos – signup *before* April 15