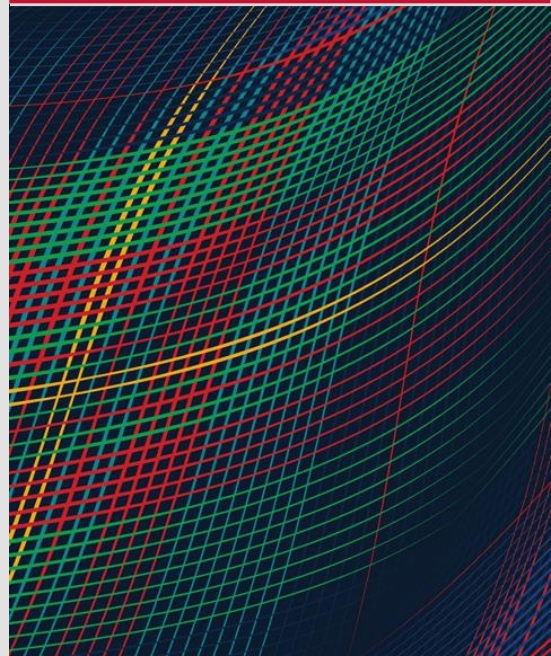
Prioritizing Work

FEBRUARY 19, 2025

Pat Place
Agile Transformation Team

Carnegie Mellon University
Software Engineering Institute



Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM25-0239

Agenda

- The “Why” of Prioritization
- Prioritization Techniques
- Estimation Techniques

Presentation Name

The “Why” of Prioritization

Carnegie Mellon University
Software Engineering Institute

Why?

Can’t work on everything at once!

Prioritization helps:

- Manage requirements and resources (including people, time, and budget)
- Manage the unknown unknowns (e.g., the addition of a new “must have” requirement
- Improve communication (by eliminating guesswork as to what is next)
- Encourage people to rethink requirements (e.g., determine some that are not really needed)
- Release software in phases (determines what will be released in each phase)

How?

Old way – assign everything into one of 5 priorities where Priority 1 is the highest, Priority 5 is the lowest. But:

- Typically assign requirements to a category
- Too many individual's "pet" requirements end up as #1
  - If everything is #1, then nothing is #1
- Rarely any further thought to prioritization

Newer way – rank order all requirements from #1 (most important) to #n (least important). But:

- Determining rank order can be hard, particularly with a lot of requirements
- Defining the meaning of importance

What?

Importance of a requirement is not an absolute value. It can depend on:

- Role; e.g., a requirement important to an admin may be of no interest to a user
- Frequency of use; someone who uses a feature infrequently is likely more tolerant of inefficiency than someone who uses it all the time
- Centrality to the mission; some requirements contribute more to the mission than others

Importance alone isn't enough; agile approaches typically use value

Principles

Prioritization is essential for any agile project

- Can’t work on everything at the same time
- Arrange work in order of value
- Allows the team to consider the minimum necessary to create value

Goal is to produce a ranked order of work items

Must be performed periodically

- New work items arrive and must be inserted into the backlog periodically
- Priority/perception of value changes as we learn

Don’t be stupid!

- Don’t ignore dependencies, if A depends on B, do B first, regardless of prioritization technique results
- Generated priorities can be adjusted

Practice

There's a lot of theory of prioritization and methods to choose between

Have to decide which method, or methods, to pick. Factors include:

- Number of items that have to be prioritized
- Frequency of reprioritization
- Prioritization team size
- Prioritization needs such as near-term vs. long-term work
- Time available for prioritization

Often find that a combination of techniques works best

Presentation Name

Prioritization Techniques

Carnegie Mellon University
Software Engineering Institute

Many Different Techniques

There are many different prioritization techniques:

- MoSCoW
- Cost of Delay & Weighted Shortest Job First (WSJF)
- Mission Based Prioritization

Other techniques include:

- Stack Ranking
- Kano model
- Opportunity Scoring
- Relative Weighting
- 100 Dollar Test/Dot Voting

MoSCoW

Pick an endpoint such as next # months or next release then divide work into:

Efficient way to sort through a large number of work items

Need to be realistic about the Must haves

Often used as a precursor to other prioritization techniques to order the work items

Cost of Delay & $WSJF_{-1}$

Cost of delay helps quantify the economic value of completing a work item sooner than later

- Focuses on estimate of revenue that will be generated by the work item
- Divide that by the estimate of how long to complete the work item
- The higher the result, the higher the priority

Revenue has to be substituted for some other estimation of value in most Government projects

Can be hard to get reasonable estimates of revenue/value

Cost of Delay & $WSJF_{-2}$

WSJF helps by using relative estimation to understand:

- Value
- Time criticality
- Risk reduction/Opportunity enablement
- Job duration/size

Then $WSJF = \frac{Value + Time\ criticality + RR/OE}{Job\ duration/size}$

Estimation of value is still an issue, but easier when using relative values

Lots of debate about the formula (e.g., should Time criticality be a multiplicand and not an addend)

Job size is easier to estimate than duration but gives less accurate results

Not easy to use with a large number of work items

Mission Based Prioritization

Choose recorded, important characteristics then determine whether a work item has/does not have that characteristic. Uses estimates of difficulty, uncertainty, and job duration as multipliers

Data for each work item is recorded, and a spreadsheet computes the priorities

Characteristics include:

- Provides strategic or tactical advantage
- Foundational (other features depend on this feature)
- Replaces deficient functionality
- High sponsorship / visibility
- Fulfills direct user request
- Required to field
- Hardens security
- Time critical / urgent

Mission Based Prioritization is typically less work when re-prioritizing – the work items still exhibit the same characteristics

Presentation Name

Estimation Techniques

Carnegie Mellon University
Software Engineering Institute

Many Different Techniques

We can’t be precise about the various factors that play into the different prioritization techniques so must come to consensus on estimates for the factors

- Use relative and not absolute estimation
- Typically use an arbitrary scale (Fibonacci sequence/T-shirt sizing)
- Don’t have to be precise – the goal is speed not precision

There a number of different estimation techniques – so choose what works for you:

- Planning Poker
- White Elephant Estimation

Other techniques include:

- T-shirt sizes/Bucket system
- Affinity Estimation
- Dot voting

Planning Poker

Each estimator has a deck of cards with numeric values

Each estimator's card is revealed simultaneously to avoid

Discussion follows to reach consensus

History:

- Based on Wideband Delphi Technique from Rand (1950/1960s)
  - Based on Delphi Technique (from 1940s)
- Created to break a deadlock in a meeting (2002)
- Became popular when published in Mike Cohn's *Agile Estimating and Planning (2005)*

Advantages of Planning Poker

1. Eliminates bias or influence
2. Brings teams together, with all voices being heard
3. Easy to use and fun
4. Relatively quick
5. Reveals gaps
6. Encourages sharing different opinions
7. Improves future work planning

Disadvantages of Planning Poker

1. Can be time consuming
2. Hard to use for large teams
3. Harder to operate in hybrid/virtual teams
4. Failure to “follow the rules” negates benefits

White Elephant

Team stand-up in a semi circle facing their sizing or white board marked with estimation values

Have deck of work item cards, in arbitrary order, in front of the board

Signal for the next member to perform one of the following steps:

- *Either take a card from the top of the deck read it out and place in one of the columns (a.k.a **propose the estimation** for that item)*
- ***OR** take one of the cards already placed on the board and move to another column (a.k.a **change the estimate**). If someone is moving card, he/she need to provide some reasons for doing it*
- ***OR** pass, if all the stories are placed and they are satisfied with the story placement*

Once all the stories are placed on the board, the team inspects the board and each member can propose to move one of the stories' place. They can, later on, discuss it with the product owner and ask questions that will help them estimate those stories together.

Advantages of White Elephant

1. Fast and engaging
2. Collective wisdom
3. Promotes discussion
4. Transparency and ownership
5. Works well for large teams/large number of items
6. Kinetic

Disadvantages of White Elephant

1. Requires active facilitator
2. Occasional instances of churn
3. Noisy

Presentation Name

Summary

Carnegie Mellon University
Software Engineering Institute

Key Points

For successful prioritization:

- Items should be well-defined with clear definitions of success
- Need more people on the team with experience than not
- Estimation technique chosen based on number of items to estimate & team size

There’s no “right” technique

- Combinations such as MoSCoW and WSJF can be highly effective

Presentation Team

Pat Place
Senior Member of Technical Staff

Telephone: +1 412.268.87746
Email: prp@sei.cmu.edu

To change the image, follow these instructions.	
Mac Instructions	PC Instructions
1. Select the placeholder image. 2. Select the Picture Image tab. 3. Choose the appropriate option (e.g., “From a File,” “This Device”). 4. Select the new image. (It will resize automatically.)	1. Right-click the placeholder image. 2. Select Change Picture. 3. Choose the appropriate option (e.g., “This Device,” “From Stock Images”). 4. Select the new image. (It will resize automatically.)

The photos should be in grayscale. To convert a color image to grayscale, follow these instructions.	
Mac Instructions	PC Instructions
1. Select the placeholder image. 2. Select the Picture Format tab. 3. Choose Color. 4. Select the rightmost image under Color Saturation.	1. Select the placeholder image. 2. Select the Picture Format tab. 3. Choose the Color dropdown. 4. Select the first image under Color Saturation.

Contact Us