HOW TO SLICE A FEATURE

1 PREPARE THE INPUT FEATURE

WARNING – Don't slice Features unless something is needed in the next PI

Does the Feature satisfy INVEST*
(Except, perhaps sized appropriately)

NO

Reformulate the Feature to clearly communicate the benefit or slice off one or more smaller Features which do satisfy INVEST and carry clear benefit.

YES

Is the Feature size less than 1/10th of your program velocity?**
(or typically medium or smaller).

You're done

Continue. You need to split it.

* INVEST Features should be:
Independent
Negotiable
Valuable
Estimable
Sized Appropriately
Testable

** Velocity varies between programs but as a rule of thumb a program should be tackling at least the 'Top 10' Features hence the no greater than 1/10th of the program velocity guideline.

2 APPLY THE SLICING PATTERNS

Does the Feature have a simple core that provides most of the benefit and / or learning? This is often the happy path with some basic error handling.

KISS (KEEP IT SIMPLE)

Could you slice the Feature to do that simple core first and build on it later with further Features?

Does the Feature include lots of optional behavior (for example different ways to achieve the same goal)?

DEFER OPTIONAL BEHAVIORS

Could you make the optional behaviors separate Features to be done once the core functionality / most popular option is in place?

Does the Feature lend itself to being released incrementally to different areas of the business?

SEPARATE BUSINESS VARIATIONS

Could you deliver it one business at a time? Could you start with the simplest business variant to generate quick wins and fast feedback?

Does the Feature need to be delivered over different channels, different mediums or different routes to the customer?

SEPARATE DIFFERENT CHANNELS

Could you deliver it one technology / one channel at a time? Could you start with the channel or most value to the business and add the other channels over time?

Does the Feature involve different user groups with different goals?

Does the Feature involve different user groups that want different sets of stories?

ADDRESS DIFFERENT USER GROUPS INDIVIDUALLY

Could you give each User Group their own Feature? This can help you to better understand the benefits to each group. See also Break Out Common Enablers.

Does the Feature involve lots of data from many sources?

Could you deliver benefit with a sub-set of the data? Could the data be consumed incrementally or sourced from existing secondary source.

CONSIDER INCREMENTALLY SOURCING DATA

Does 80% of the value come from 20% of the Stories?

FIND A STORY GROUP

Could you find the set of most valuable Stories and develop and release them as their own Feature?

Do many Features rely on the same underlying system behaviors (often making the first of them selected to be very large and complex)?

BREAK OUT COMMON ENABLERS

Could you break out the common enablers into their own 'Architectural' Features? Delivering the enablers can significantly de-risk, simplify and reduce the estimates for the other, related Features.

Does the Feature include Special Variations?

ISOLATE SPECIAL VARIATIONS

Could you focus on the most popular / highest volume cases first and treat the more specialized corner cases as separate Features? You may find that their value / cost ratio is very small and they are never needed.

LAST RESORT

BREAK OUT A SPIKE

Are you still baffled about how to slice the Feature?

Can you define the 1..3 questions most holding you back?

Write a Spike / Knowledge Enabler with those questions in mind.

Do the minimum to answer the questions and then start again at the top of this process.

3 EVALUATE THE SLICING

Are the new Features roughly equal in size?

NO

Try another pattern on the original Feature or the new Features that are too large.

YES

Do each of the new Features readily fit into a PI (< 1/10 of the program velocity)?

Try another pattern on the original Feature or the new Features that are too large.

Do each of the new Features satisfy INVEST?

Try another pattern to see if you can get this.

Is there an obvious Feature to start with that gets you early benefit, learning, risk reduction etc?

Try another pattern to see if you can get this.

You're done, though you could try another pattern to see if it gets better results.

WARNING DON'T:
- **Defer non-functional requirements**
- **Slice too early**
- **Over slice**
- **Slice by component**
- **Forget the Feature testing**

IVAR JACOBSON INTERNATIONAL

ivarjacobson.com

Inspired by, and complementary to, "How To Split A Story", Richard Lawrence, www.agileforall.com