

# CONNECTING VALUE STREAMS AND FEATURE SETS

How to keep million-dollar software developments  
from getting out of control

BRIAN LOOMIS, CHIEF ARCHITECT, PROGRESS SOFTWARE

NOVEMBER 11<sup>th</sup>, 2021



An Association for All IT Architects

**“Two things that remain eternally true and complement each other, in my view are: don’t snuff out your inspiration and power of imagination, don’t become a slave to the model; and the other, take a model and study it, for otherwise your inspiration won’t take on material form.”**

**~ Vincent VanGogh (November 1882)**

# Agenda topics

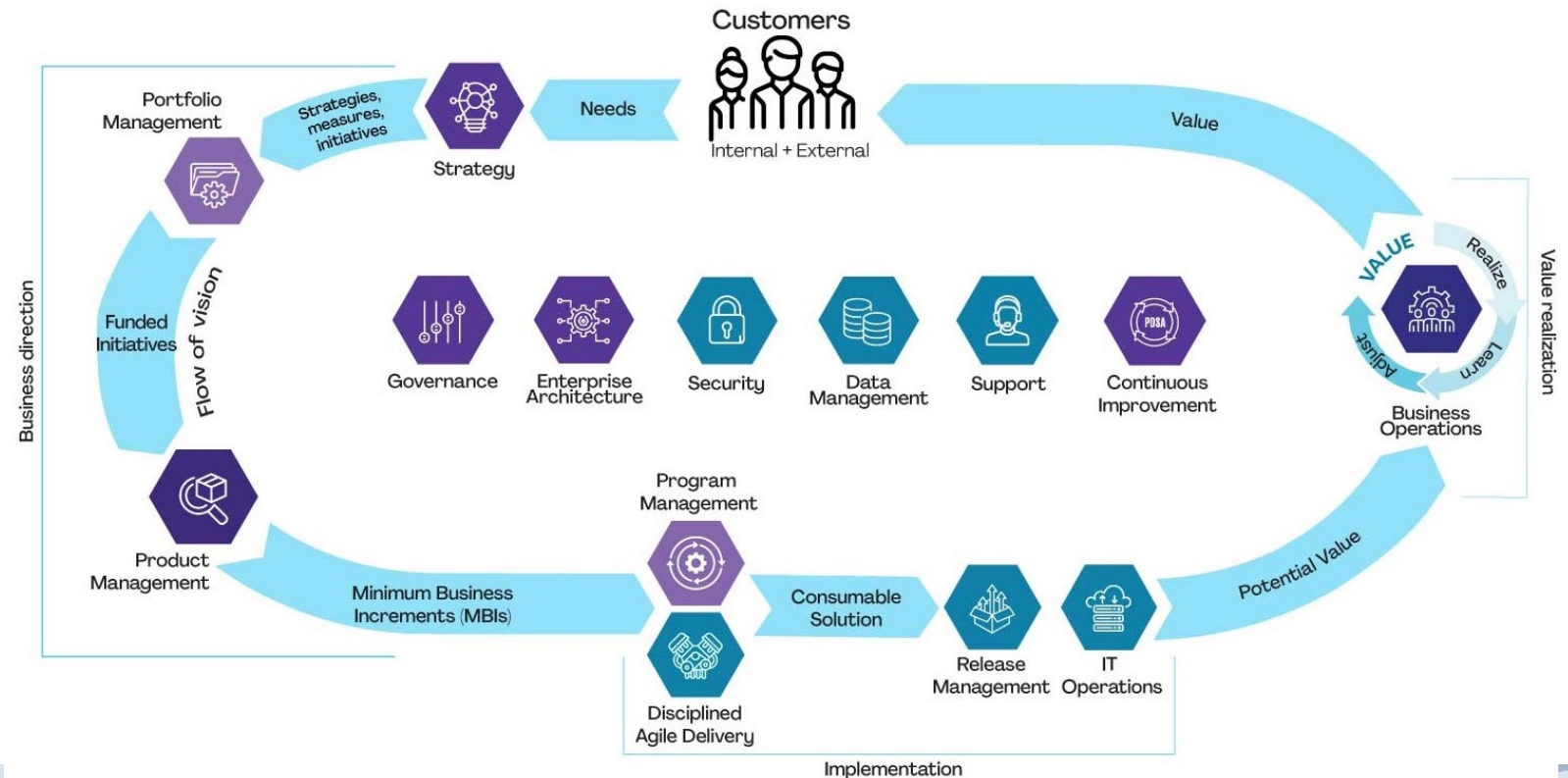
- Hoarding v. refactoring – do you find the discussion a bit old?
- Do you see the same patterns in growing team, then rapidly downscaling, then trying to pick up the pieces?
- Inconsistent investment strategy

# How does PI or sprint planning go?

- Fixed resources (how many resources/hours can we allocate?)
- Fixed schedule
- Leads to: accepting functionality
- “I have two 2x4s, what can I build?” (even if I don’t need them)


# Value streams historically

- “A value stream comprises all people and systems necessary to deliver value and the associated flow of information and materials. It results from the sequence of individual steps that are necessary to deliver value.”



# What are value streams in SAFe?

- Operational value streams = business processes (or loosely Lean/BMC)
- Development value streams = SDLC processes (define, build, test, release)
  
- Neither one answers the questions:
  - What do I actually get in terms of value in *my* process? (from customer)
  - When is this development done, how many sprints? (from CFO)

- 
- Why is this so hard?
    - External value is hard to measure
    - # sprints is calculated by when our value stops... well, when is that?
    - Most software efforts are really just too big to fail, or we throw resources at it until-successful
  - There will come a time when the answers to these questions matter
    - It may matter to you as an architect, picking up legacy spaghetti (and decisions/short-cuts)
    - It may matter to you as a CIO (when the insurance policy for a known data breach is the same as buying land in Malibu)
    - It may matter to you as a CEO, when startups cannibalize your business model

# What if the conversation were...?

- “I have two 2x4’s but can go to the hardware store for anything I want, including a rocket ship?”
- Bring value into the sprint plan
- Each story has an estimate
  - Of resources required
  - Of value (and of corresponding risk)\*
- We used to call this earned value



# Portfolio planning

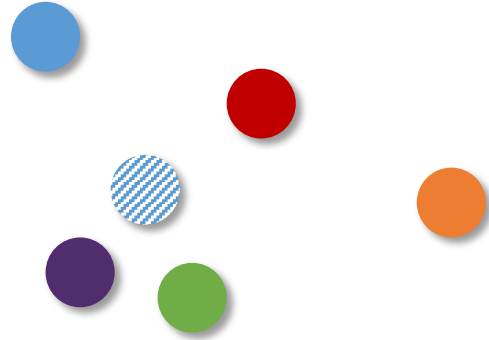
- How do you invest your own money?
- Lots of buckets, each with different expected payoff, whole set is optimized for a potential return
- What if we looked at a software product and said, “we want it to generate \$5MM by end of year (knowing we’re at \$1MM right now)”
  - What if we started at 4.9MM current state?
  - Obviously it would shape our risk tolerance and the actual investments

# So, what are the software investment types?

- To get new customer, new persona/capability → blue ocean
- To upgrade existing customers, new capability → incremental revenue
- To reduce churn, not lose customers → revenue consistency
  
- Scenario-driven → new persona, aggregating
- Transformational/structural → platforming, productizing, SaaS/cloud migration
- Integrations → partners
- Functional/product → new feature sets (how deep to go? best of breed?)
- Quality → ease of use/barrier to entry, performance (could include investment in documentation, in-app guidance, etc.)
- Tech debt reduction → efficiencies (esp. in operations)
- Marketing, sales discounts/offers, customer support → other differentiators (also competing investment choices to engineering investment)

# By stakeholder... how does it affect value? Risk?

- Enterprise standards
- customer reported
- partners
- corporate alignment
- field (cust arch)
- field (account team or cust support)
- product roadmap (PM) & analysts
- open source / community
- arch prototype
- maintenance
- refactoring
- tech debt



# By project style, there is an investment style

- MVP compared to enterprise development
- Balanced portfolio vs. new feature heavy/experimental
- “Just barely good enough”
  
- Be skeptical when everyone around you is doing a very different style (many/most orgs cannot do MVP because they have not assessed value or their risk tolerance, which is why they end up being MINO's)

# How do customers pick your product?



# Where does refactoring come in?

- ...
- And dependencies often need to be bundled, sequenced to calculate true value/cost (and avoid leaving out the un-sexy database schema effort that will take 4 months)



# How do I do people planning?



# What does a 3RR look like?



For further information, please [contactus@iasaglobal.org](mailto:contactus@iasaglobal.org)