

Professional Scrum with UX (PSU) Synopsis

<https://www.scrum.org/courses/professional-scrumtm-user-experience-training>



'Got 57/60 questions correct. Lengthy third-party practice exams helped. 99% of it was Scrum fundamentals (assuming you know them). Agile Coaches and Scrum Masters should review these short Scrum exams with their teams [Example 1] [Example 2] ...

Topline. This was a fairly general, high-level, and abstract course on how to shoehorn traditional multi-year UX activities into 2-week Scrum Sprints for SMALL teams and projects (how to decompose complex tasks into smaller chunks)—With a focus on basic Lean UX tools (with the assumption Scrum teams are NOT required to augment basic 2-week Scrum Sprints with independent Product Management or Lean UX groups)—**It didn't take much imagination to understand how this applied to other long-lead item disciplines** (enterprise architecture, business requirements, testing, security, hardware, manufacturing, contracts, strategic planning, procurement, other general business functions, etc.)—This was NOT a course on Scrum SCALING, Product Management, nor Lean-UX (Scrum is strangely silent on these disciplines)—However, the trainer emphasized Scrum allows augmentation with other principles, practices, and techniques (i.e., "*The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory*")—Our trainer was reticent to discuss extracurricular practices commonly associated with Scrum (i.e., Epics, Features, User Stories, Acceptance Criteria, Story Points, Estimating, Story Mapping, TDD, BDD, CI, CD, DevOps, etc. because they're not based on Kanban manufacturing principles)—[That being said, software development is knowledge work and not well suited for Kanban manufacturing models according to experts such as Christoph Roser](#)—Our trainer also steered the class towards software development applications of Scrum, although the Scrum guide does not define Scrum as a software development method in order to promote Scrum for non-software applications (Kanban trainers are everywhere and they want to force Scrum into a tiny software box so they can reserve non-software projects for Kanban)—Again, many trainers specialize in Kanban but exploit Scrum's popularity to get new Kanban clients (in other words it pains them greatly to have to grit their teeth and teach something in which they do not believe in order to bait-n-switch Kanban for Scrum)—**The most I learned from this class was a reinforcement of modern, basic Scrum principles, how to shoehorn large work items into two-week Sprints, and some exposure to a smattering of Lean-UX techniques** (and, of course, how much Kanban trainers despise Scrum) ...

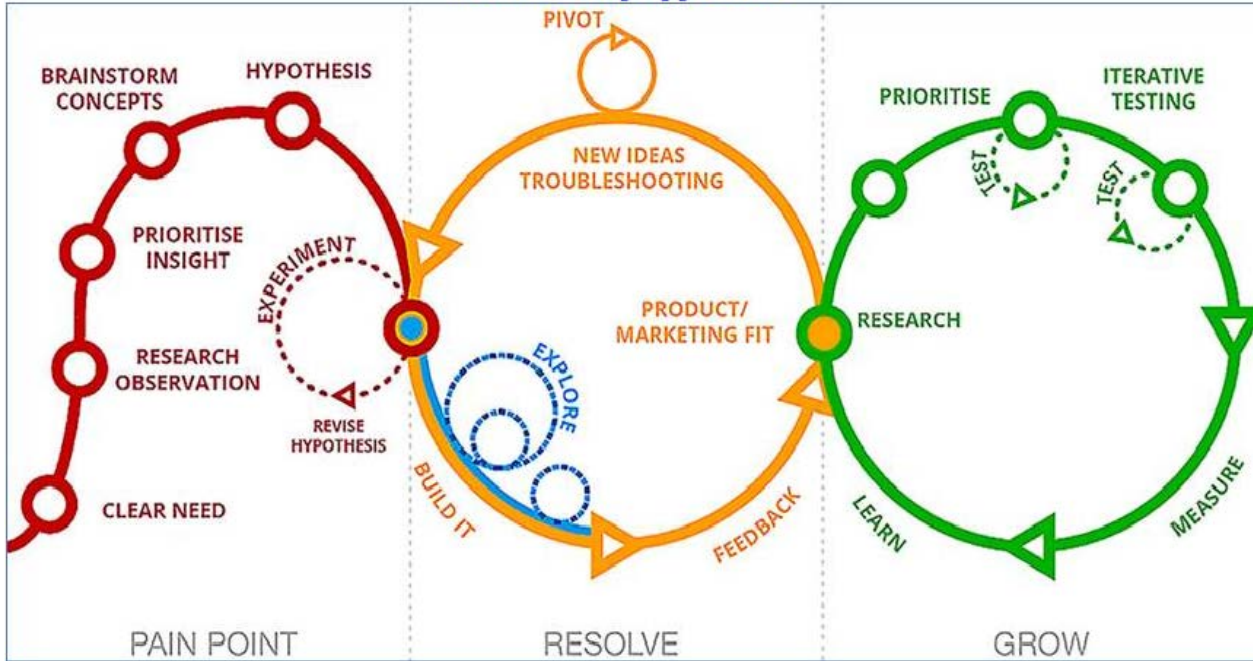


- * The instructor was a very intelligent, knowledgeable, and highly talented British Scrum, Kanban, and UX trainer.
- * He employed all of the usual suspects (i.e., Zoom, Slack, Mural, and a variety of other training tips, tricks, and techniques).
- * It was a small class made up of UX and Scrum experts from across Europe (mostly UX folks wanting to learn about Scrum).
- * (User Experience) UX is a big upfront traditional linear waterfall product analysis process (these were NOT "Lean" UX folks).
- * UX is a lengthy process with hundreds of techniques designed to overwhelm projects with analysis paralysis (like Six Sigma).
- * It's important to note that the term "UX" is a bit dated and is branching out into other areas (user interaction, user research, etc.).
- * This course is designed to hand jam traditional UX into a two-week Scrum sprint, fusion, or hybrid (you can imagine everyone's horror).
- * Not even SAFe attempts to smash UX into a couple of days of a two-week Scrum sprint (and even Lean UX simply ain't that friggin lean).
- * He began by a short lecture on the [latest Scrum Guide](#) and emphasized its rather diluted, innocuous, and vague concepts it has recently evolved.
- * This had a 4 fold purpose (i.e., review Scrum basics, prepare us for the course exam, emphasize Scrum's extensibility, and open door for Kanban).
- * He queried each of us on Scrum's ambiguous boundaries like a court prosecutor (framing Scrum as a "[software method](#)" which its guide doesn't do.)
- * The [latest Scrum guide](#) only uses the word software once and is clearly framed as a general-purpose product and service development approach.
- * This was probably the trainer's Kanban bias as the Kanban community's claim to fame is domain independence (framing agile as only for software).
- * The trainer tormented us on Scrum's ambiguities, suggested Scrummasters may go away, and skipped over the Product Owner role as irrelevant.
- * Then he switched gears and dove into Lean UX values, principles, and practices, although he said this was NOT a course on Lean UX which it was.
- * Clearly, this was NOT a full-blown UX course, which would require years to teach with its bloated Six Sigma-like bag of tips, tricks, and techniques.
- * He walked us through a short case study, although his career UX trainees clearly were beyond any practical business experience or commonsense.
- * He'd ask for business goals and they'd respond with "conduct extended UX research" otherwise business goals could not properly be established.
- * He covered about 127/219 slides in four hours through the use of simple Mural team exercises, which I enjoyed immensely (and frequent breaks).
- * Again, he insisted that UX is NOT a separate Product Management group and all (traditional) UX should be jammed into two-week Scrum sprints.
- * His intent was to introduce "[Dual Track Scrum Sprints](#)" as a technique for performing alternating UX and development Sprints by a combined team.
- * In his early Scrum Guide interrogation, he emphasized "[Scrum is purposely incomplete](#)" so he could jam UX and [dual track development](#) into Scrum.
- * We use "[Dual Track Development](#)" out of necessity, although gov't contracts discourage its use (as teams should be coding vs. designing/discovering).
- * Scrum teams are viewed as computer programming teams so they should always code at maximum velocity (and everyone else isn't subject to Agile).
- * The "[Scaled Agile Framework](#)" also has "[Dual Track Development](#)" built into it in the form of Lean-Agile Solution/Product Management and UX teams.
- * Projects do SAFe [Solution](#) and [Product](#) Management very poorly or not at all (forcing Scrum teams to do "[Dual Track Development](#)" under the radar).
- * My personal goal was to continue mastering how to properly integrate [Solution](#) and [Product](#) Management, [Design Thinking](#), and [Lean UX](#) into SAFe.
- * The Scrum line is for teams to absorb UNFUNDED/UNBUDGETED [Solution/Product](#) Management, [Design Thinking](#), and [Lean UX](#) (without slowing down).
- * All of this nonsense is starting to make sense to me after 2 years of intense analysis, research, and experience which is what I wanted out of this course.
- * Our trainer ignored Lean Startup, Startup Way, Design Sprints, Product Management, Lean UX, Business Experiments, and other competing paradigms.
- * I'm already a [Certified SAFe Agile Product Manager](#) and read dozens of books on lean, agile, and traditional product management (but I'm still learning).
- * SAFe's [Agile Product Manager](#) forms Value Stream and Product Market Analyses, UX Designs, Visions, Strategies, Roadmaps, Valuations, and Measures.
- * In some sense, I feel as though Lean UX encompasses/compresses SAFe's Agile Product Management, which can't be done in a sprint or two (or can it).
- * Perhaps SAFe doesn't clearly integrate these concepts as cleanly as it should, which is the source of my confusion (and everyone else's for that matter).
- * Projects tend to do the "easy" SAFe things like form Scrum software development teams (and every other abstract SAFe best practice falls by the wayside).
- * Traditional product managers are allergic to Lean, Agile, Lean Startup, Startup Way, Design Sprints, Lean UX, Experimentation, Scrum, Kanban, SAFe, etc.
- * The 2nd day was tense as the instructor was loaded for bear and keyed off on us a few times which is normal and my WIFI cut in and out (but got through).
- * Trainer admitted a Scrum Sprint may be an experiment vs. product (which he sheepishly admitted because Kanban is all about quickly manufacturing widgets).
- * Kanban community hasn't realized new product development evolved into a minimal set of rapid Lean UX experiments (and diverged from Lean manufacturing).

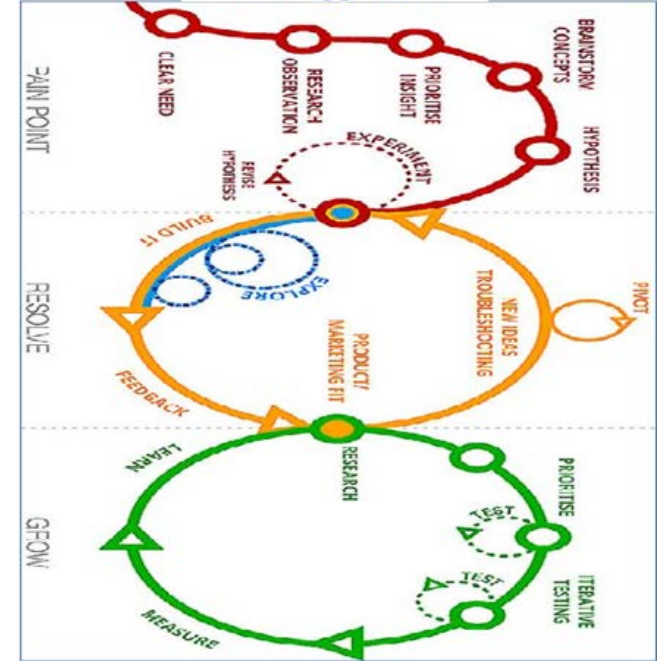
Footnote 1. Our Scrum UX trainer seemed oblivious, immune, and/or unwilling to suggest a multi-phase business or project framework like the one below (maybe even in overlapping cycles which SAFe embraces)—Maybe our trainer didn't like Eric Ries (Lean Startup, Startup Way) or Jake Knapp (Design Sprints) which is probably the case (or these uniquely American approaches are threatening to Kanban which was popularized by Europeans) ...

Footnote 2. See my new [Agile Product Management whitepaper](#) to help untangle the morass of Agile Product Management (APM) frameworks and how to successfully integrate LeanUX principles and practices into Lean-Agile frameworks without eating the cost of LeanUX by slowing agile teams down ...

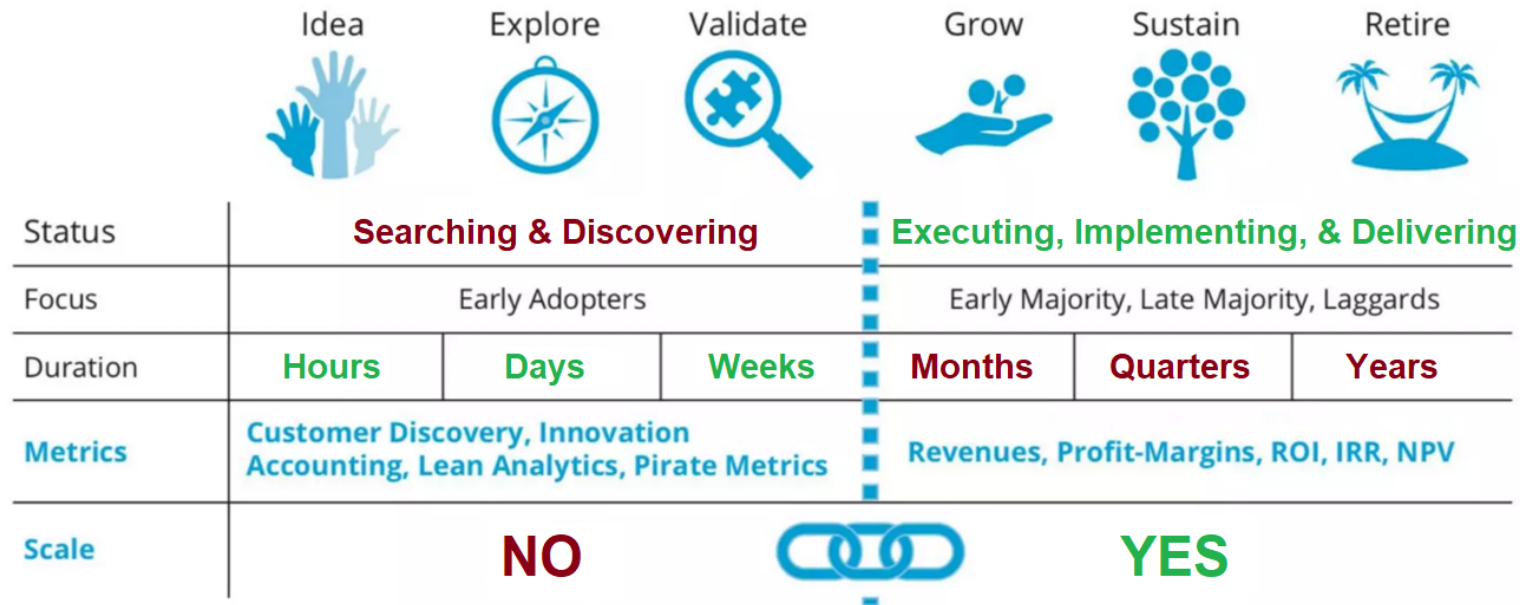
Lean Startup Approach



SAFe Approach



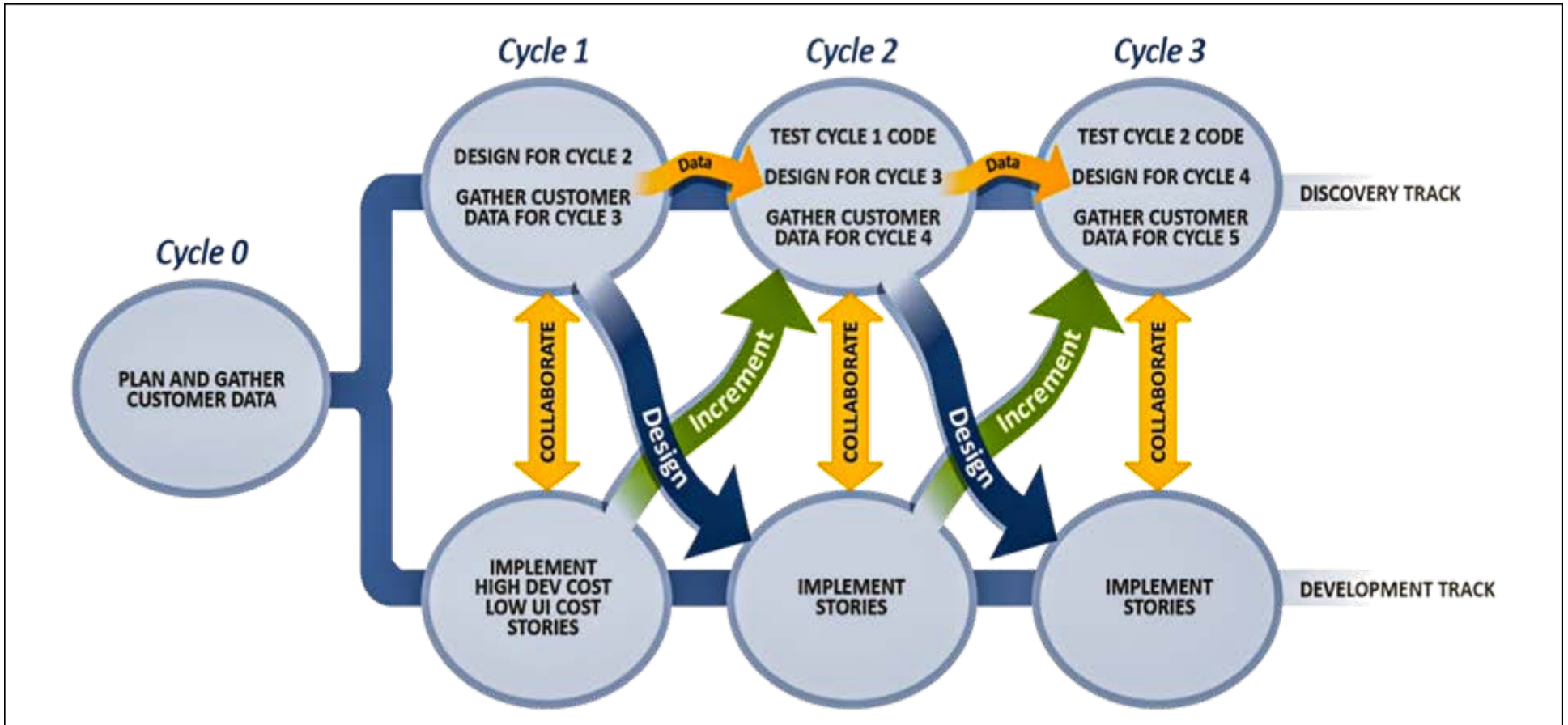
Agile Product Lifecycle



Viki, T., Strong, C., & Kresojevic, S. (2021). [Lean product lifecycle: A playbook for making products people want](#). Harlow, UK: Pearson.

The **TOP THREE** bubbles **BELOW** in the Dual Track Scrum Development Chart are called **DESIGN vs. DISCOVERY** (the latter of which is not well liked in U.S. government acquisitions for some) ...

The two rows or lanes can either be two parallel teams (design and development) OR Scrum teams can simply alternate design and development sprints (either way you're gonna double your value and halve your velocity)—Personally, I believe Lean-Agile Startup, (Product) Design Sprint, Product/Solution Management, Design Thinking, and UX should be separate Scrum teams feeding development teams just-in-time UX designs (in other words, Scrum is a framework that should be used to encapsulate ALL of these frameworks regardless of team, phase, or function)—In the lean world these are called CONWIP systems like SAFe (to differentiate themselves from double-loop Kanban systems) ...



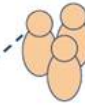
Our instructor is recommended a single team alternating between or hybridizing design and development (from sprint-to-sprint)...

Cross-functional Discovery & Delivery Team = The Scrum Team

Discovery

Discovery & Delivery

Delivery

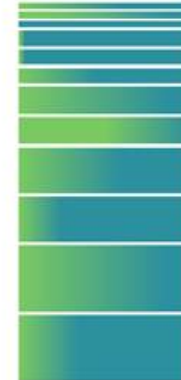
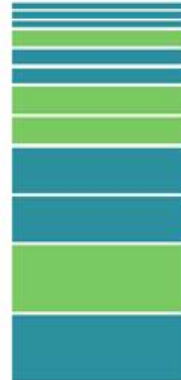
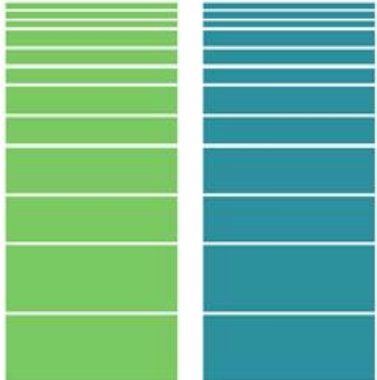


- Two types of work, one team moving together as a unit
- Non-sequential execution is key
- Everyone pitches in regardless of individual skill specialty

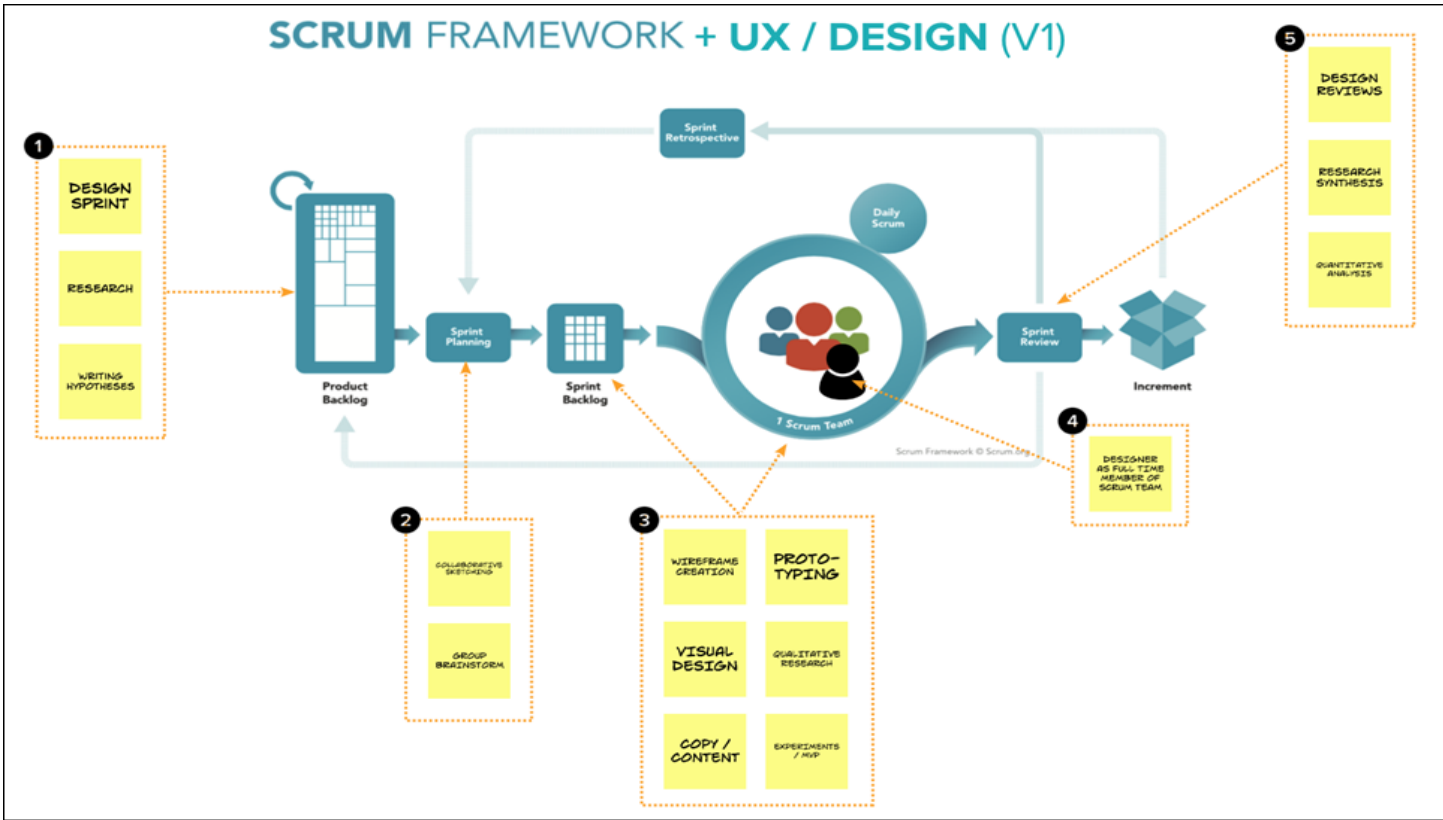
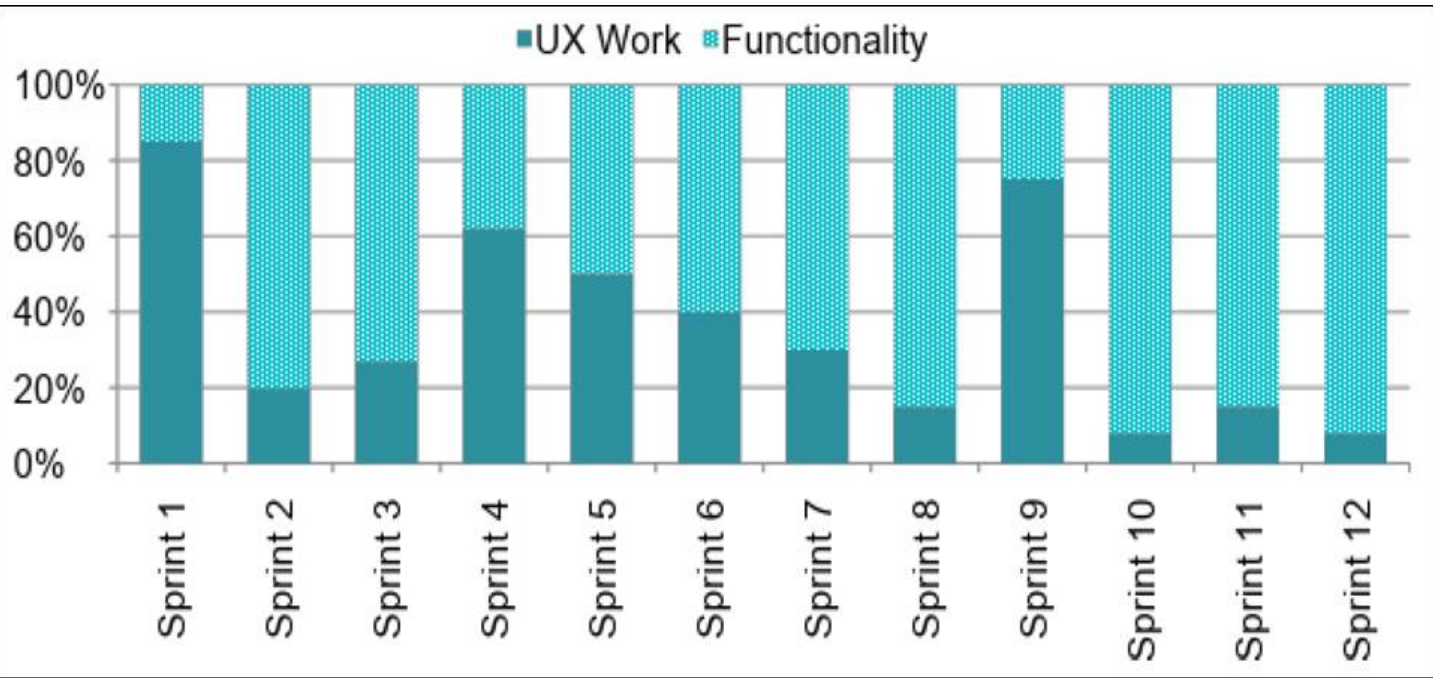
Separate Backlogs

One Product Backlog
UX Work **AS** PBIs

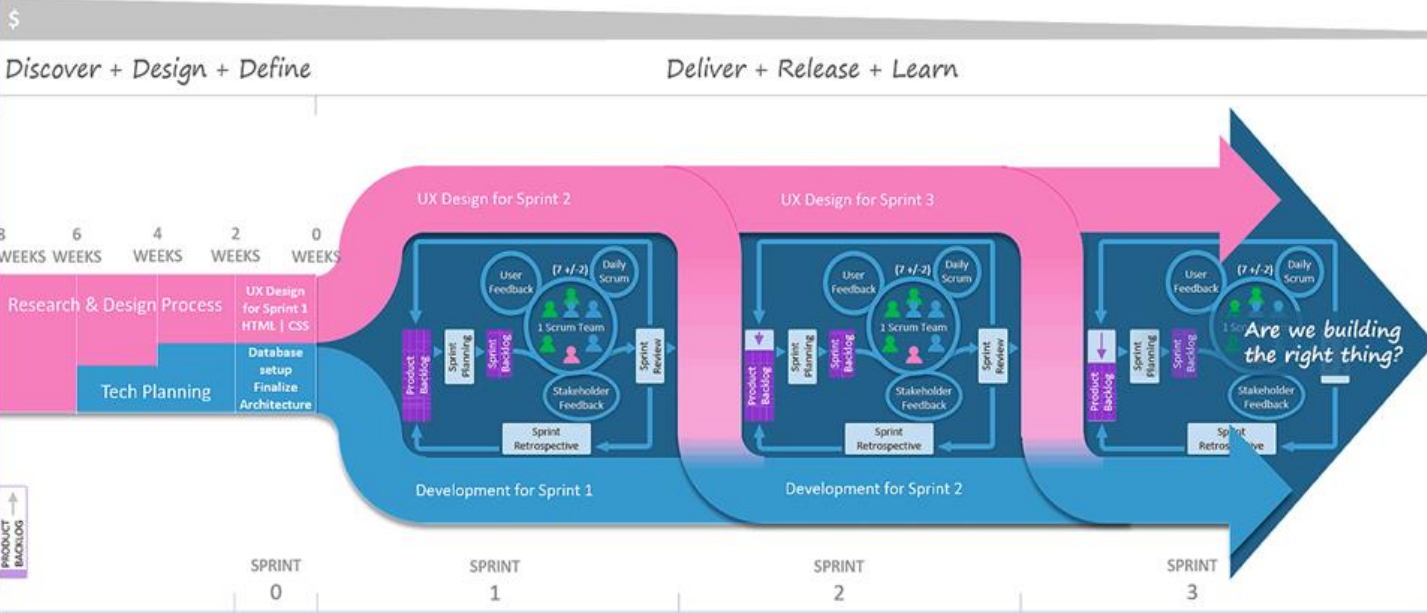
One Product Backlog
UX Work **IN** PBIs



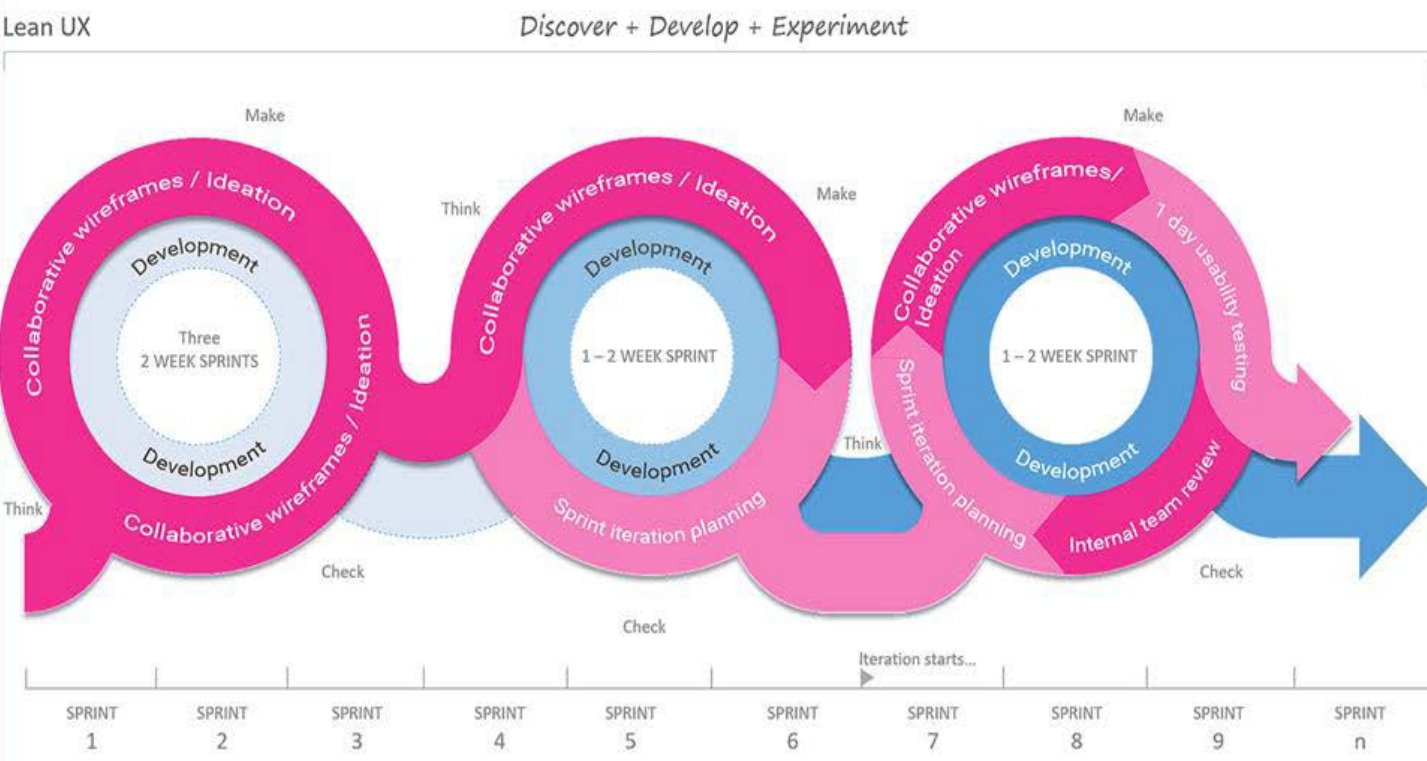
UX DELIVERY



Agile UX

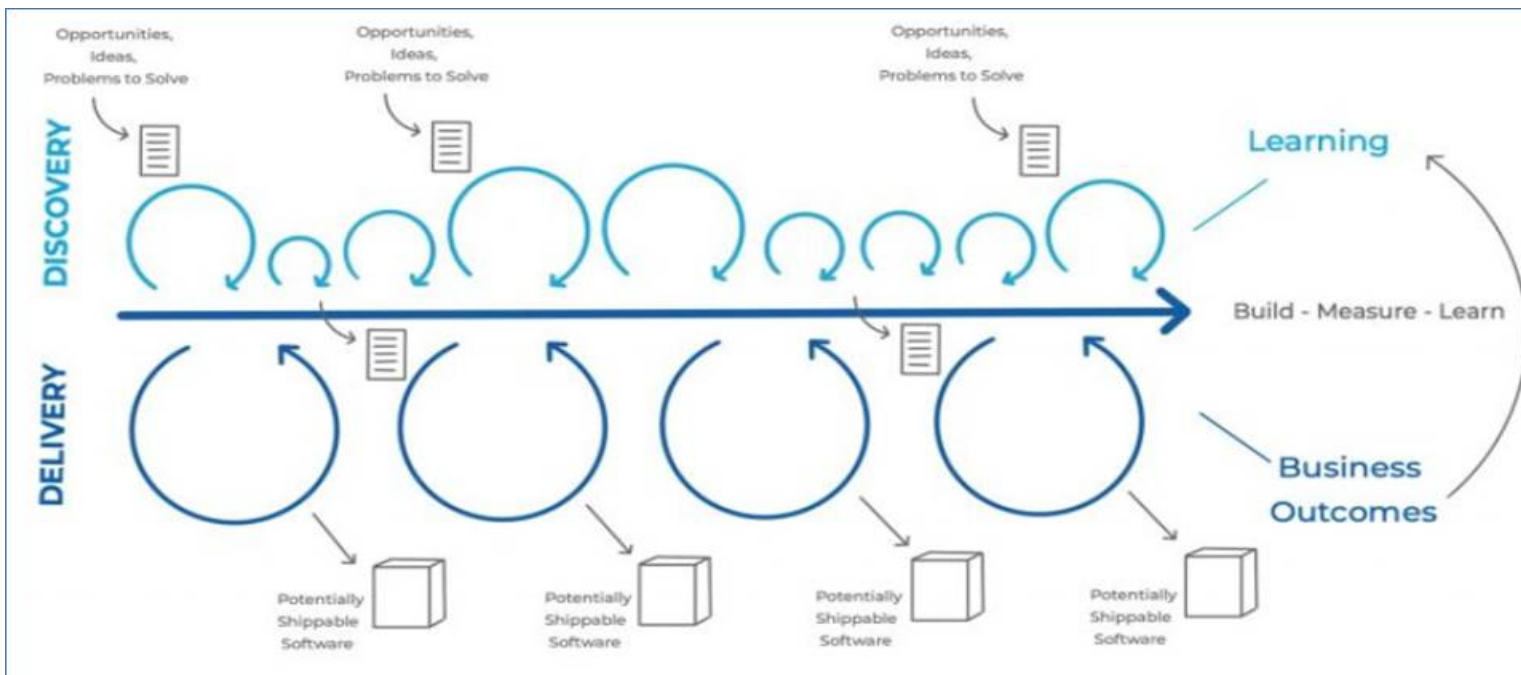
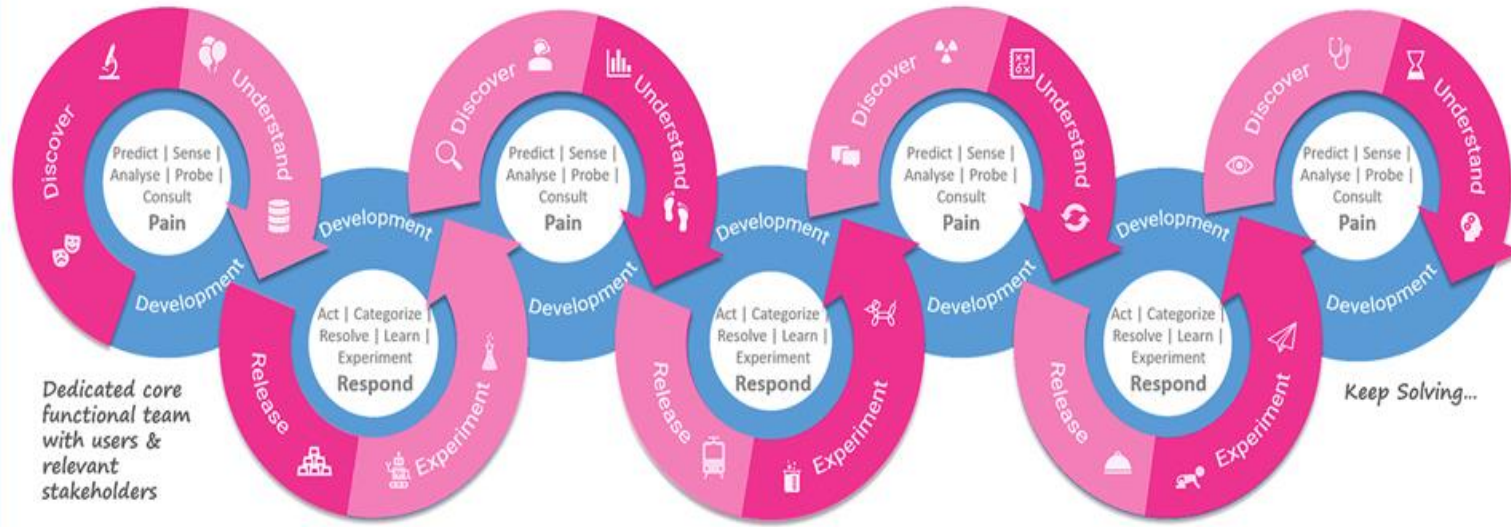


Lean UX



Pain Driven Design

\$



Footnote: Some interesting articles on Dual Track Agile Product Development ...

- <http://jpattonassociates.com/dual-track-development>
- <http://uxpajournal.org/adapting-usability-investigations-for-agile-user-centered-design/>
- <http://uxdesign.cc/dual-track-agile-is-a-game-changer-for-designers-e19705fec3c1>
- <http://aktiasolutions.com/dual-track-agile>