The Essence of Software Startups

• A practical tool for software startups
  / "Survival kit" aimed especially at early-stage startups
  / Provides essential practices and tools for tracking progress

• Based on the Essence Theory of Software Engineering

• Based on scientific research but focus highly on the practicality of the artifact
What the software system must do to address the opportunity and satisfy the stakeholders.

The requirements:
- Establish a shared understanding of what the software system must do.
- Communicate the intent of the software system to be produced.
- Define the capabilities, services and qualities that the stakeholders desire from the system.
- Are organized to allow the scope of the software system to be managed.
- Drive the development and testing of the system.

Software System

Way of Working

Involved

- Representatives assist the team.
- Timely feedback and decisions provided.
- Changes promptly communicated.

Ready

- User documentation available.
- System accepted as fit-for-purpose.
- Stakeholders want the system.
- Operational support in place.

Working Well

- Predictable progress being made.
- Practices naturally applied.
- Tools naturally support way-of-working.
- Continuously tuned.

Conceived

Bounded

Coherent

Acceptable

Addressed

from: http://semat.org/quick-reference-guide
The Essence Theory of Software Engineering

• A method-agnostic software engineering tool for project use
  / Progress management
  / Method engineering
  / Encourages communication and learning

• NOT a methodology or a model
  / More akin to UML than e.g. SCRUM
    ➔ In fact SCRUM can be modeled using Essence
The Essence Theory of Software Engineering

• Consists of
  / Kernel – contains elements present in every SE endeavor
  / Language – used to extend the kernel
• Kernel has three views
  / Alpha view
  / Competency view
  / Activity view
• Each view divided into three areas
  / Customer
  / Endeavor
  / Solution
The Essence Kernel Alphas
The Essence Alphas

- Progress management
  - Project management (not comprehensive)
- Each alpha has states
  - Depicts progress from project start to finish

The Essence Theory of Software Engineering

• Language is the defining feature
• A modeling tool that provides guidance for method engineering and carrying out a project
• Without extending the kernel, Essence is very general or ”bare bones”
• Complemented by a practice library from where one can pick practices to use
Problems with Essence

• Resource-intensive to adopt
• Lacks good learning resources
  / You have to read a book to get into it
• Suffers from current lack of practitioner interest
• Does not offer much practical advice, relies on the user to extend it
  / Too much "Google this for more info" type content in e.g. practice cards
And what about those startups?
Essence for inexperienced developers?

• We studied 100+ student teams using Essence in a project-based SE course

• These students found Essence useful because of their inexperience
  / Helped guide them into the right direction
  / Encouraged them to reflect on why they work the way they do
  / Provided ideas for how to work (practice library)
  / Helped them track progress on their project

• Startuppers are also often inexperienced -> maybe Essence could help them too?
The Essence for Software Startups?

- Startups engineer software differently
  - Ad hoc, various agile practices; highly varied
  - Technical debt
- Startups also need to work systematically
- Startups are very focused on learning
- Based on our experiences with students, early-stage (student) startups often feel lost on what to do at the earlier stages
- But…
  - Startups are not SE projects, they are firms
  - Essence is not very practical as is
The Essence of Software Startups

- Alphas are progress management tools -> closely related to metrics
  - We are studying software startup metrics
- Based on data so far:
  - New business alphas for the kernel
  - New area of concern for the kernel: business
- Focus on PRACTICE; activities and activity spaces
  - Currently focused on revamped cards
  - Similar to the practice library of Essence; "startup essentials" (e.g. vs. their "Agile essentials")
  - Cards provide guidelines on what to do at the earlier stages
  - Currently tested in the Lean Startups course of 2018 at JYU
  - Early-stage software startups
The (Software) Startup Cards

• Based on existing research (mostly by the network), and practitioner literature
• Information mapping: every card has the same layout
  / Motivation
  / What to do
  / Common mistakes
  / (Sources)
• More Essence-inspired than Essence
  / No kernel or language so far
• Cards seem to be useful so far?
Data Collection

• How to evaluate this?
• Current plans
  / Observation, mentoring; tracking progress and how they use the cards
    → What metrics to actually focus on?
  / Interviews / survey after course?
    → Potentially the best and simplest way to measure if it was useful is to ask directly…
Current State?

- VERY early testing
- Cards are not the theory, merely a synthesis of existing research and practitioner literature
- The kernel and the language are left out for now; alphas are not a focus
  / Maybe later in the course?
The Endgame?

• Is this just another consulting card deck?
• How to make this actual theory \textbf{while also} keeping it practical?
• This is a PhD
  
  / Most value for the academia comes from the papers comprising it
  ➔ Metrics papers etc.
  
  / The artifact being designed here is iteratively created using the Design Science Research Methodology (DSRM) for Information Systems
  ➔ This is just the very beginning
References