fluent@agile - the game

The game is a platform for teams to move towards becoming a high impact team by helping them understand where they are, where they are striving to go, and by giving the team a strong feeling of ownership of their journey.

This document provides a guide for facilitating the game and material for printing and building the game.

Christian Vikström, Peter Antman (Spotify/Crisp) 2014

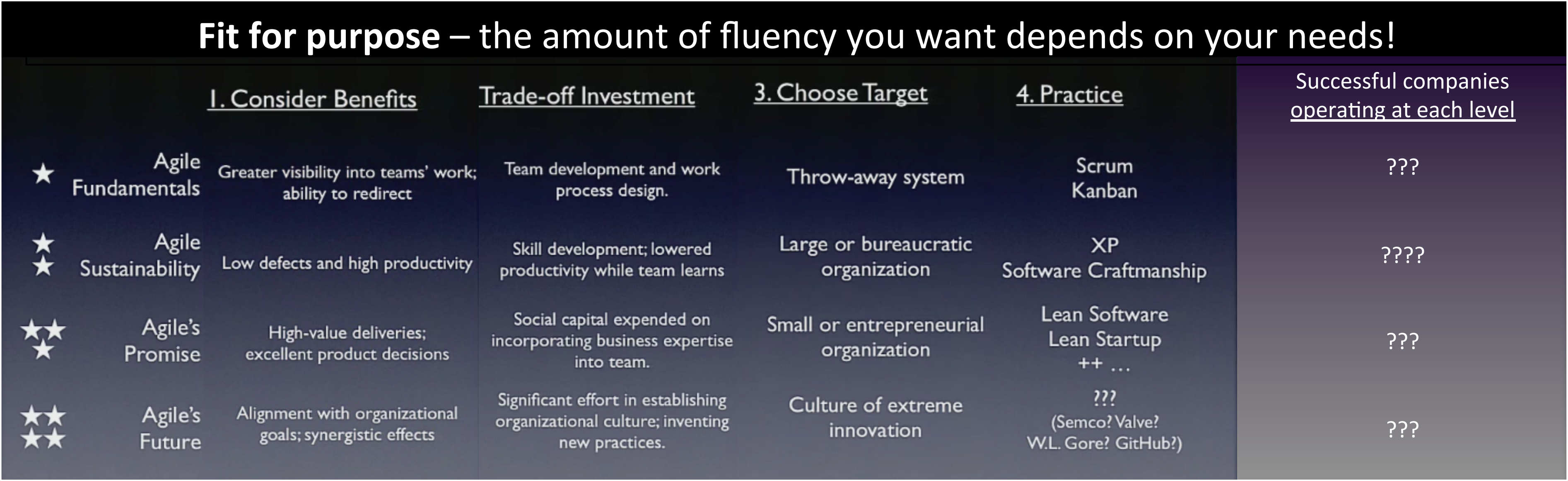
christian.vikstrom@spotify.com
peter.antman@crisp.se

Three pillars for the game

- **Based on a model:** The game is based on a model for how teams usually moves on their journey from “pre-agile” to high impact agile teams.
- **Visualization:** We believe that visualizing is a super strong way to build shared understanding, engagement and collaboration.
- **Ownership:** The game is built so that teams can take a strong ownership over their journey towards becoming a high impacting agile team.

Agile Fluency*

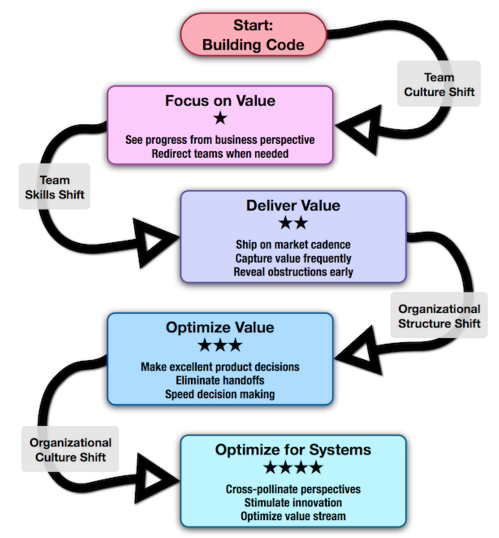
**The fluent@agile game is heavily based on the "Your Path through Agile Fluency" by Diana Larsen and James Shore*



Fit for purpose – the amount of fluency you want depends on your needs!

		l. Consider Benefits	Trade-off Investment	3. Choose Target	4. Practice	Successful companies operating at each level
★	Agile Fundamentals	Greater visibility into teams' work; ability to redirect	Team development and work process design.	Throw-away system	Scrum Kanban	???
★★	Agile Sustainability	Low defects and high productivity	Skill development; lowered productivity while team learns	Large or bureaucratic organization	XP Software Craftmanship	????
★★★	Agile's Promise	High-value deliveries; excellent product decisions	Social capital expended on incorporating business expertise into team.	Small or entrepreneurial organization	Lean Software Lean Startup ++ ...	???
★★★★	Agile's Future	Alignment with organizational goals; synergistic effects	Significant effort in establishing organizational culture; inventing new practices.	Culture of extreme innovation	??? (Semco? Valve? W.L. Gore? GitHub?)	???

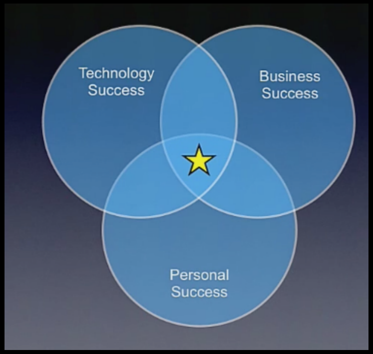
A Team's Path Through Agile Fluency



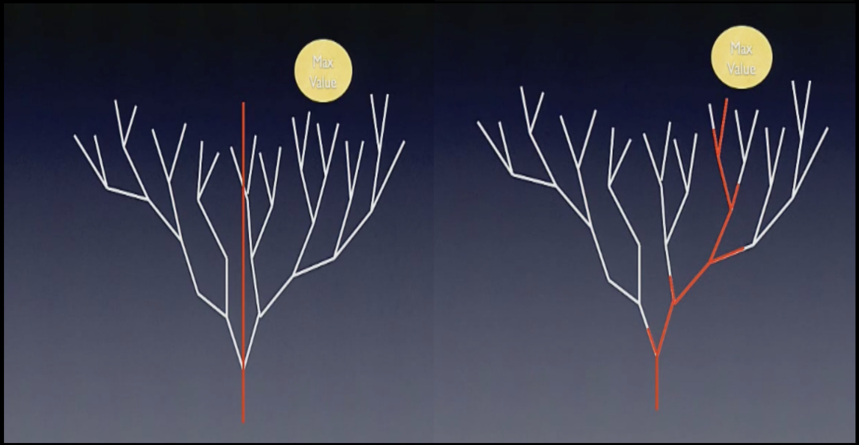
Good Business **MAKES** the Company

Bad Technology **BREAKS** the Company

People **GLUE** the Company Together



Work the Plan **Learn & Adjust**



Proficiency vs. Fluency:

Fluency is unconscious competence; how well you do under pressure

Fluency is how a team develops software when it's under pressure. Anyone can follow a set of practices when given time to focus in a classroom; true fluency is a skillful, routine practice that persists when your mind is distracted with other things.

For Agile, we're considering ***team fluency*** rather than individual or organizational fluency. Agile development is fundamentally a team effort, and your organization's success with Agile will depend on the fluency of your teams.

Team fluency also depends on management structures, relationships, organizational culture, and more.

Don't make the mistake of blaming individuals for low team fluency, or assuming that one highly-skilled individual will guarantee high team fluency

Choose your level of fluency

We've seen that teams progress faster when they practice advanced techniques alongside basic techniques. Teams' practices become more deeply and reliably grounded when they work this way. So, it's best to choose the level of fluency you want to achieve and to practice everything needed for that level from the beginning.

Different approaches to facilitate

We believe that one of the pillars for the game to be successful is that the team feels a strong ownership over the game and their improvement work. Our experience is that when we facilitate in a way where we listen to the team and try to adapt our facilitation along the way makes it easier for the team to take that ownership.

Possible ways to facilitate:

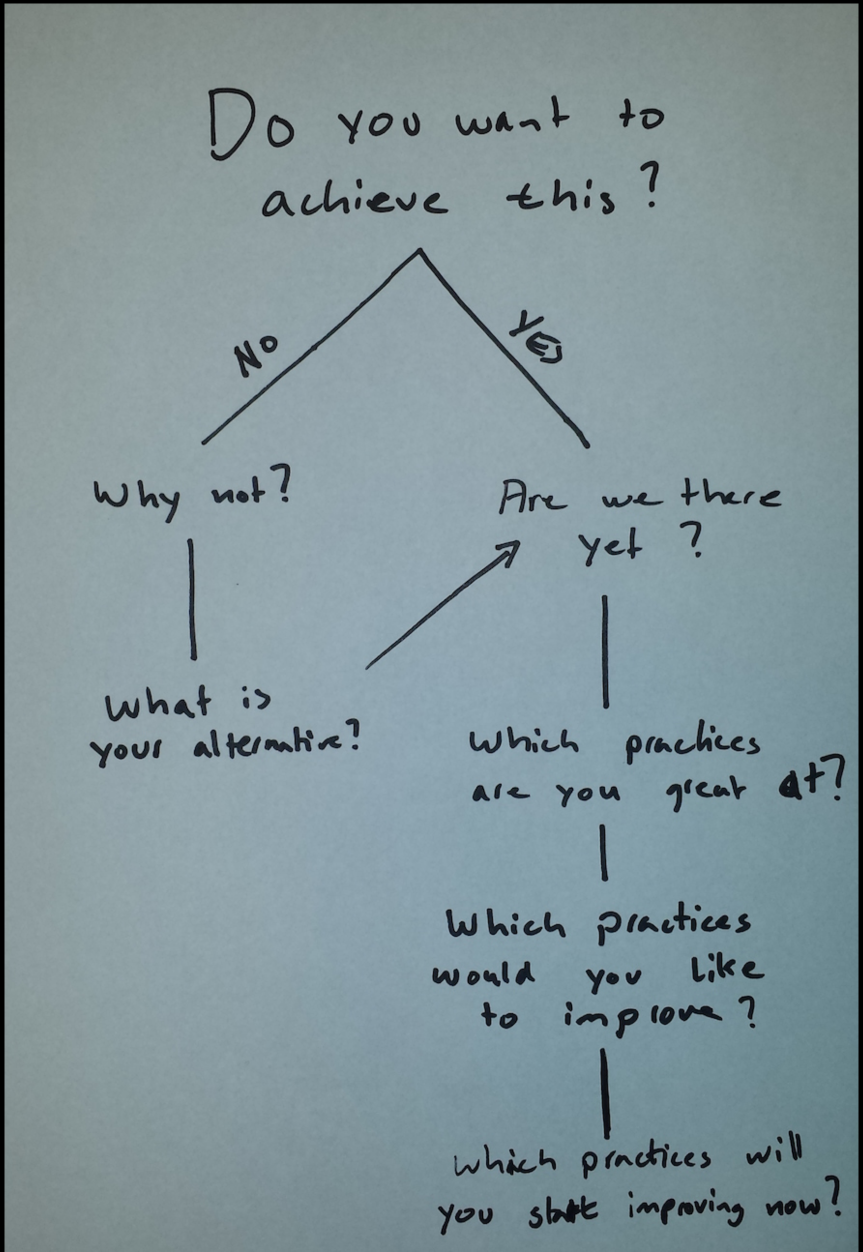
1. **Present the model approach:** Present the model/game and let the team build the game and then decide on what practices they want to work on getting fluent on next. Establish the value/goals of implementing the practices. Write one or several improvement stories as first steps towards getting fluent at the practices.
2. **Music metaphor approach**: Present the music metaphor. Either by presenting it yourself or by facilitating a discussion that builds the metaphor. Let the team(s) build their models using a training from the back of the room approach.
3. **Training from the back of the room approach:** Give the team minimal instructions to get started with the game, stay available for the team to pull information or give direction if the team gets stuck or misunderstand how to play the game.

Present the Model approach

- Introduction to the model/game.
  - Put the different levels up on the wall. Describe shortly what each level means, stay open to answer questions or have short discussions with the team on each level.
- Team discussion: “At what level in the model do you as a team want to get fluent?”
  - “Are you fluent on any of the levels already?”
  - “Where would you like to be as a squad?”
  - “On a personal level, where do you think you would be most satisfied?”
  - If the team seems open to it you might even go for a consensus decision on this.
- “What practices do you believe you need to be fluent at to be able to be fluent at all the levels up to the level you are aiming at?”
  - Let the team go through all the practices notes and discuss which ones they are: (“teaching moments” often arise here when team members ask what specific practices are, and what they are good for :)
    - Fluent at (put them on the road)
    - Practicing (put them besides the road)
    - Need to do (put them under the “drivers license”)
    - Don’t need (put them under “Park”)
  - “Are there any practices missing in the game that you are already doing or that you think you need to be doing?”
  - “Is there any practices that we can remove and still get fluent?”
- Next practice(s) to get fluent at:
  - Create a target condition:
    - Let the team pick something they want to start working on to get fluent at, e.g. by dot-voting.
    - Discuss and define what it would mean for the team if they got fluent at the practice(s). E.g. by brainstorming post-its on the possible benefits, then discuss and group them; now you have a goal/target condition.
  - Write improvement stories that describes the first steps to take to move towards fluency.
- How to integrate this improvement work to your daily work:
  - Discuss how the team will make this happen as part of their daily work. E.g.:
    - The improvements stories are prioritized together with the rest of their work in the backlog.
    - Having weekly meetings to plan/groom their improvement work.
- Optional perspectives:
  - “Given the pace of improvement work we are currently having in the team - how long do you think that it would take for your team to get fluent at all the practices that you have listed?””
  - “Do you believe that it would be valuable for your team and for the organization to increase the pace of improvement work?”

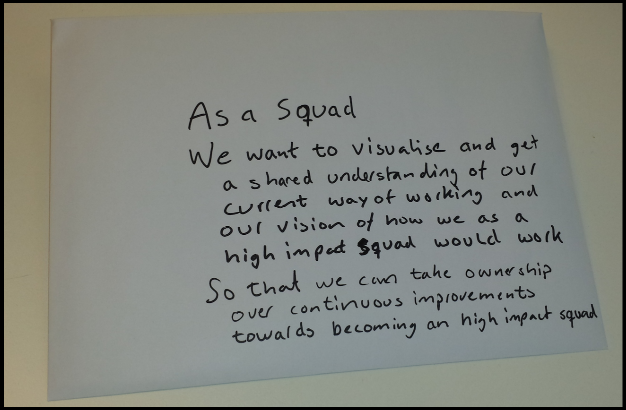
Training from the back of the room approach

- Do a really short presentation of the level(s) you want to focus on and then give this sheet of questions to direct them during the workshop


Do you want to achieve this?
No
Yes
Why not?
What is your alternative?
Are we there yet?
Which practices are you great at?
Which practices would you like to improve?
Which practices will you start improving now?

Preparations for facilitating

- Print the Game(s) (one for each squad) (see below on how to do this). Put in an envelope and write the “user story” on it, or just write the squad name.
- Make sure you have:
  - enough häftmassa (the sticky stuff you put on the back of the notes to stick them to a wall) if you will build the game on the wall/whiteboard
  - Post-its of the right color and size; so that the team can write new “practices notes” (i.e. orange, yellow, green & pink); an alternative is that they write the new practices on blank practice notes
  - Sharpies
  - Scissors (unless you cut the papers before hand)
  - Optional: A movable board that you can bring from your workshop to your squad area (with your built game on :).



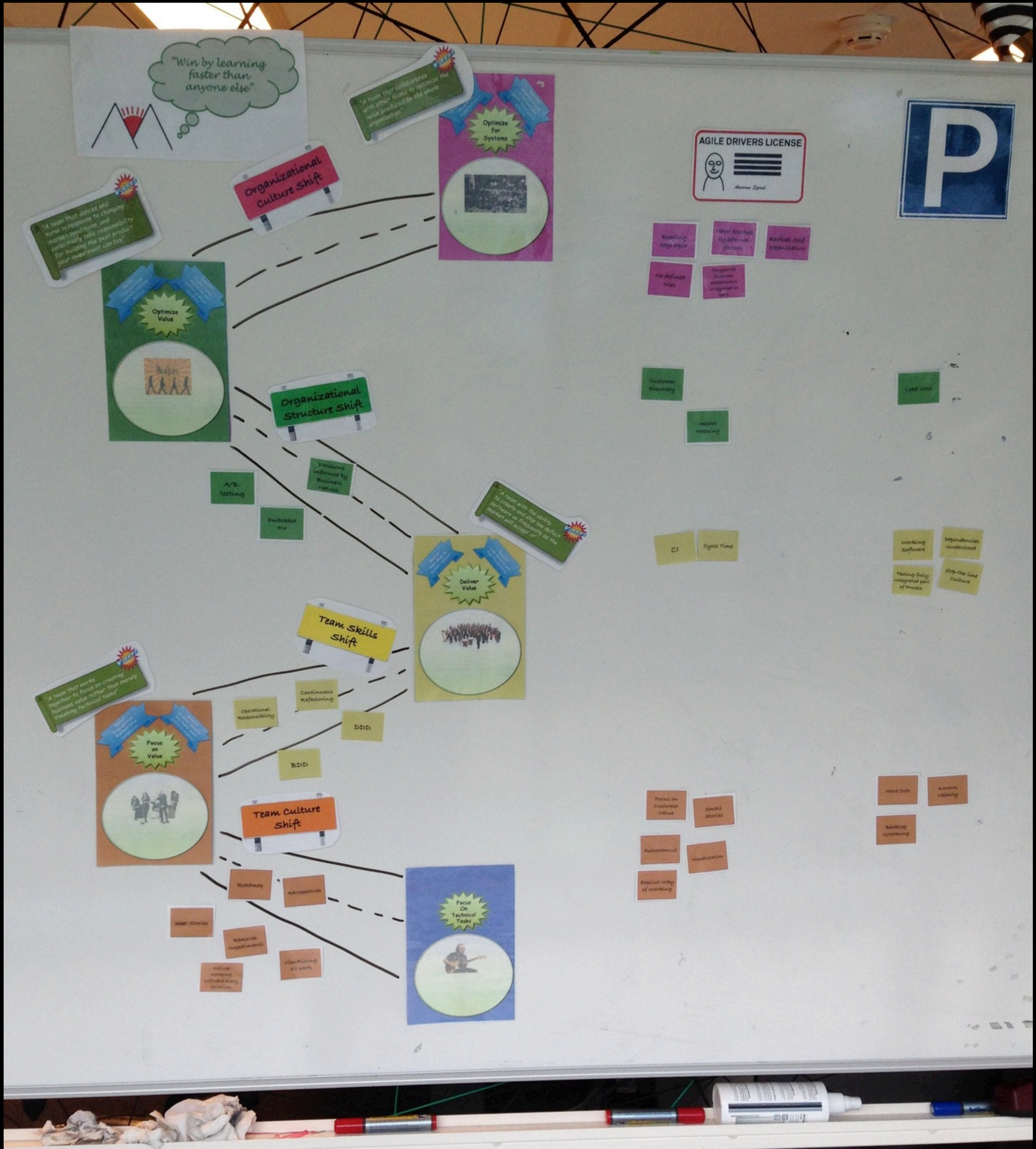
As a Squad
We want to visualise and get
a shared understanding of our
current way of working and
our vision of how we as a
high impact Squad would work
So that we can take ownership
over continuous improvements
towards becoming an high impact squad

Envelope to team

“User Story”

As a Squad
We want to visualize and get a shared understanding of our current way of working and our vision of how we as a high impact team would work
So that we can take ownership over continuous improvements towards becoming an high impact team

An example on how a game could look like for a team


"Win by learning faster than anyone else"
"A team that collaborates with other teams to optimize the value produced by the whole organization"
Optimize for Systems
Organizational Culture Shift
"A team that dances and turns in response to changing market conditions and collectively take responsibility for building the best product your investment can buy"
Optimize Value
Beatles
Organizational Structure Shift
A/B testing
Decisions informed by Business Metrics
Embedded PM
"A team with the ability to create and ship low defect software as frequently as the market will accept it"
Deliver Value
Team Skills Shift
"A team that works together to focus on creating business value rather than merely finishing technical tasks"
Focus on Value
Operational Responsibility
Continuous Refactoring
DDD
BDD
Team Culture Shift
Roadmap
Retrospectives
User Stories
Removes Impediments
Visualizing all work
Deliver working software every iteration
Focus On Technical Tasks
AGILE DRIVERS LICENSE
Awesome Squad
Bleeding edge agile
Never blocked by external factors
Radical self organization
No defined roles
Customer Discovery
Impact Mapping
Lead time
CI
Cycle Time
Working Software
Dependencies understood
Testing fully integrated part of Process
Stop-the-line Culture
Focus on Customer Value
Small Stories
Autonomous
Visualization
Explicit Way of Working
Have DoD
Known Velocity
Backlog Grooming
P

See more:

- Our talk at Agile Sverige (in swedish)
- Fluent_agile-as2014.pptx

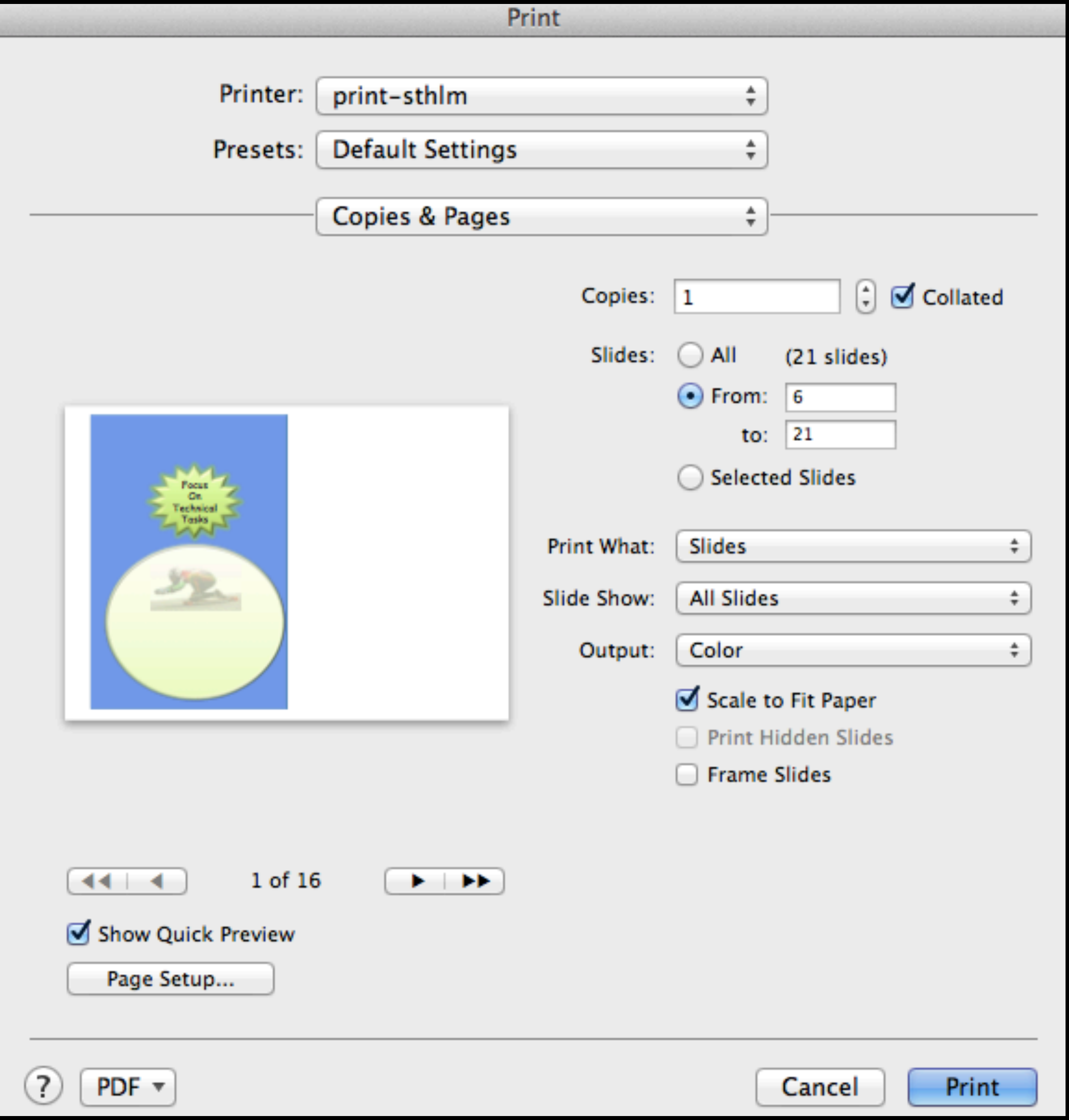
Building the game

- Print the game; one copy for each team (see instructions below)
- Laminate the sheets (there is a lamination machine on 10th floor in the printer room*)
- Put the sheets in an envelope

** Spotify Stockholm office specific info* ☺

Building the game:

Printing



Print

Printer: print-sthlm

Presets: Default Settings

Copies & Pages

Copies: 1 ☑ Collated

Slides: ○ All (21 slides)

● From: 6

to: 21

○ Selected Slides

Print What: Slides

Slide Show: All Slides

Output: Color

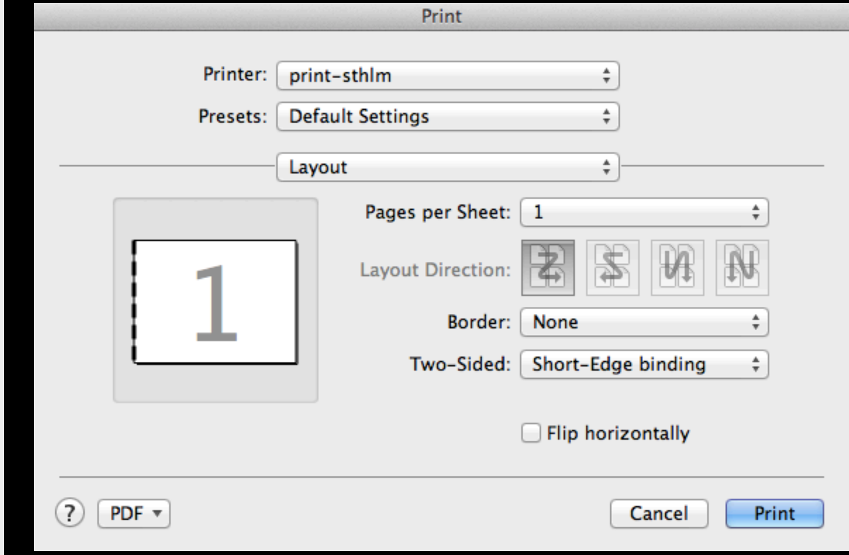
☑ Scale to Fit Paper

□ Print Hidden Slides

□ Frame Slides

1 of 16

☑ Show Quick Preview



Print

Printer: print-sthlm

Presets: Default Settings

Layout

Pages per Sheet: 1

Layout Direction:

Border: None

Two-Sided: Short-Edge binding

□ Flip horizontally

Make sure to:

- Scale to fit paper
- Two-sided: Short-edge binding
- Set the range of slides to print (first slide is the "Focus on technical tasks" slide)

Printing starts after this slide !!

Focus On Technical Tasks

Pre-Agile

Individuals or loosely coupled teams that work in an ad hoc manner or following a plan.

This state is often characterized by having a PMs, Gantt charts, detailed and comprehensive specs. Or it can be very ad hoc and reacting on stakeholder needs.

This state might resemble the “Optimize Value” or the “Optimize for System” states.

Metric:
Progress from Business Value perspective

Benefit:
Greater visibility into teams' work; ability to redirect

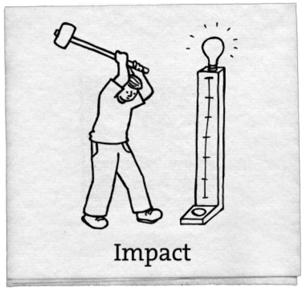
Focus on Value

Metric:
Team ships on market cadence

Benefit:
Low defects and high Productivity

Deliver Value

Agile Sustainability

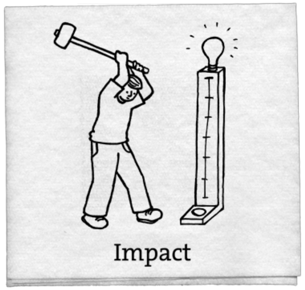


You'll see much **higher quality** software and dramatically **improved responsiveness**, shipped as often as the market will bear.

To reach fluency at the this level, study and **practice techniques** such as those described by Extreme Programming, Software Craftsmanship, DevOps, and Agile software quality gurus. Some, such as test-driven development, are of the *"moments to learn, lifetime to master"* variety.

Developing team members' skills to the point of fluency takes time and significant effort, often **3 – 24 month**. Productivity will often decrease as the team learns new skills and **pays off technical debt** in existing code.

Agile Fundamentals



You'll have **greater visibility** into what your teams are working on, and you'll be **able to direct** them towards the 20% of the work that provides 80% of the value.

Team members must learn to plan in terms of **business results** rather than technology, and they must learn to **take responsibility** for the success of the **whole team** rather than their contribution as individuals. Scrum and Kanban are methods used at this level.

It takes **2 – 6 months** of practice to shift from independent individual contributors to a **collaborative, team-based workplace**. The team may need more attention from the organization in terms of information about business priorities and customer value as well.

Metric:
Team provides concrete business metrics

Benefit:
Higher value deliveries and better product decisions

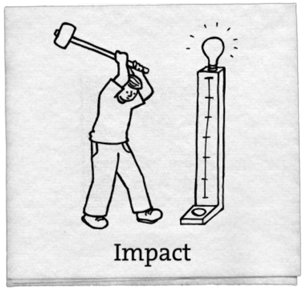
Optimize Value

Metric:
Team reports how its actions impact the overall organization

Benefit:
Alignment with organizational goals; synergistic effects

Optimize for Systems

Agile’s Future

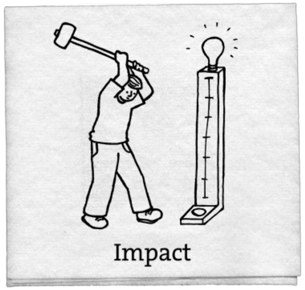


Teams contribute to **enterprise-wide success**. Team members understand organizational priorities and business direction and will sacrifice their own needs to support the needs of a product more critical to business success.

Requires shifting organizational culture to focus on the whole system, it also requires working at the bleeding edge of Agile practice and potentially inventing new ways of applying systems thinking to Agile.

For most organizations, this level of fluency is probably best left as an aspiration for the future.
However, if you’ve done the work to build and support multiple teams with solid three-star fluency, you may be most of the way there.

The Promise of Agile



The team understand what the market wants, what your business needs, and how to meet those needs. Or, as in a startup environment, they **know what they need to learn** and how to go about learning it.

It requires teams to **incorporate business experts** as full-time team members.
Lean Startup, Lean Software Development are useful. Agile chartering, embedded product management teams, customer discovery, and adaptive planning are all examples of techniques used.

Achieving this level of fluency often takes **several years**--not because of the skills required, but because people in the **organization must learn to trust the team** and its use of Agile before making changes that affect their power, control, and familiar ways of working.

2 star

VALUE added

"A team with the ability to create and ship low defect software as frequently as the market will accept it"

1 star

VALUE added

"A team that works together to focus on creating business value rather than merely finishing technical tasks"

3 star

VALUE added

"A team that dances and turns in response to changing market conditions, and collectively take responsibility for building the best product your investment can buy"

4 star

VALUE added

"A team that collaborates with other teams to optimize the value produced by the whole organization"

Team Culture Shift

Organizational Structure Shift

Team Skills Shift

Organizational Culture Shift

User Stories	Small Stories	PO has product vision	PO In team	Transparent
Focus on Customer Value	Roadmap	Prioritization of work is clear	Product Backlog	Regular Planning
Autonomous	Backlog Grooming	Have DoD	Known Velocity	Visualizing all work
Estimation by team	Trust	Good Daily Standups	Iterations	Burndown Chart
Share Responsibility	Self-organizing	Cross-functional	Delivers end-to-end	Demo

Deliver working software every iteration

Visualization

Retrospectives

Removes impediments

Explicit Way of Working

Usable Mission

Team size max 9

Scrum

Kanban

WIP limits

Co-located

Operational Responsibility	Dependencies Understood	Design Guidelines	Continuous Improvement	Stable Throughput
Cycle Time	Working Software	Easy to on-board new members	DRY	Collective Code Ownership
Continuous Refactoring	stop-the-line Culture	CD	XP	Evolutionary Design
BDD	DDD	Pair Programming	Code Review	Monitoring of systems
CI	Testing fully integrated part of Process	Zero Bugs Culture	Dev Ops	TDD

Post-mortems

Anyone in the team can deploy

No branches

Automatic Deploy

Feature Flags

YAGNI

Collective Quality Ownership

Unit Testing

System Testing

Managing Tech Debt

No Warnings in Builds

Mob Code Review

Lead time	Lean Startup	Embedded PM	Impact Mapping	User Story Maps
A/B-testing	Customer Discovery	Hypothesis	Decisions informed by Business Metrics	UX & Design is Squad Responsibility
Direction based on Learning	Automatic Rollback	Key Designs validated with real Users	Mob Programming	Innovation Accounting
Monitoring User Behavior	Testing in Production	Validated Learning	No Estimates	Design in Squad

Radical Self-organization	Tangential business stakeholders integrated in team	From concept to cash, beyond squad	Adapt to the need of the system	Value stream analysis on whole system
Never blocked by external factors	No defined roles	Bleeding edge agile		

AGILE DRIVERS LICENSE

Awesome Squad

*Please update the vision statement in this slide to fit your organization. Or as an alternative have a discussion in the team about what they see as their vision.

Hobby musician / Studio artist / Solo artist

A musician that plays by herself.

Could be a hobby musician or a professional studio musician or a solo artist

The personal skill level can span from novice to world class...

Band

Creating music **together** simultaneously

Play **others** music.

Play what the **audience want** them to play.

Have a repertoire.

Usually plays at weddings and after-ski or dansband.

Can be very successful;
e.g. ???...

Symphony orchestra

Super skilled musicians

Playing by notes

Directed by a conductor

Successful band in control of their music

In control of their music.

They write, produce and record it.

Might even have their own record company.

Delivering music that people like.

Improvising (symphony) orchestra creating fantastic music in the moment

THE LONDON IMPROVISERS ORCHESTRA

No conductor.

No notes.

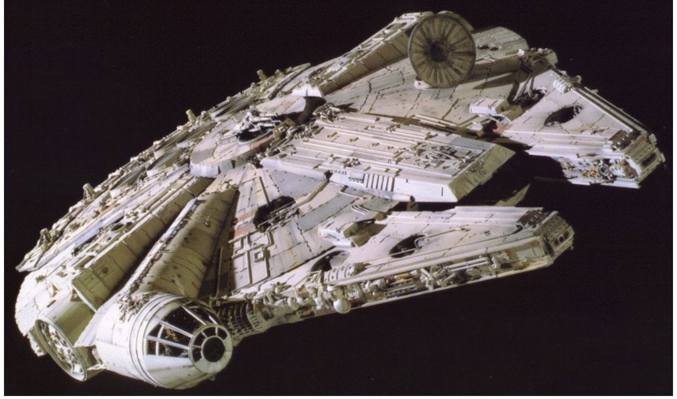
Can use signals and signs to give direction.

Every performance is unique.

Printing Stops before this slide !!

Slides with material not yet used or thrown away (i.e. ignore them...)

SAS
Scandinavian



BFGoodrich
Radial T/A

ES-NW 38H

Spare pics