Design your agile organization using (SOA) Service-Oriented Architecture Principles

Catherine Louis & Raj Mudhar
Agile2016, Atlanta, GA

wtdom.org @rmudhar @catherinelouis
Today’s Agenda

1. Introduction - 5 mins
2. WHY SOA for teams - 10 mins
3. How to model video example and instructions -10 mins
4. Prepare to model (Lego) - 5 mins
5. Model & test round 1 - 15 mins
6. Model & test round 2 (with team operations) -15 mins
7. Debrief / Discuss -10 mins
8. Close - 5 mins
What are your org design challenges?

Introduce yourself to your neighbor
Ask about their challenges building an agile org
Common Org Design Challenges

- Scarce, rare skills that have a long learning curve
- Org designed around budget vs customer need
- Product too big and broad for full-stack Scrum teams
The DSP Team Story:
https://www.youtube.com/watch?v=iB7JOsSUXO8
Are you familiar with SOA?
What is SOA?

**Services-Oriented Architecture** is an architectural style supporting services using a design approach of scheduled and on-demand requests and responses for users of the service. A service:

- is self-contained
- is a "black box" to consumers of the service
- might be composed of other services
- has a business need driving it
Psst! ... (SOT Design Principles are in the trifold on your table!)

SOT Design Principles

- Communication
- Independence
- Reusability
- Autonomy
- Statelessness
- Discoverability
- Composability
- Granularity
- Normalization
- Optimization
- Relevance
- Location transparency

Interpreting the Principles

- Adhere to a communications agreement
- Maintain relationships that minimize dependencies
- Grow knowledge, maintain relevance, be reusable
- Self-organize, with domain autonomy, able to decide how to work
- Minimize effort to manage state information: exercise minimum viable reporting
- Supplemented with communication meta-data, teams are discoverable
- Each team member adds value regardless of team size
- Teams are sized appropriately for the work they do
- Use patterns, minimally cross-team redundancy, unless bottlenecks exist - then break the rules
- Quality is preferred: teams learn and improve
- A team's purpose is obvious, no ambiguity
- Teams real-time accessible regardless of geography

Team Operations

Typical operations between teams involve 4 specific actions:
- GET - Acquire something from a team (e.g. velocity)
- PUT - Change something existing within a team (e.g. update a backlog item, evolve DoD)
- POST - Add something new to a team (e.g. work)
- DELETE - Remove something from the team (e.g. backlog items)
Modeling Basics

- Identify skills
- Compose teams
- Annotate your model
**Problem:** Not able to demo or deliver

**What gets modeled:** The Full Stack Scrum Team

**Interfaces** Primarily with: End users

---

**Problem:** Limited, rare experts. Not enough to go around

**What gets modeled:** A Service team - DSP example

**Interfaces** primarily with: Other full stack teams, other service teams

---

**Problem:** Not enough work for any single resource on a team

**What gets modeled:** A specialty pool

**Interfaces** Full Stack Scrum Teams

---

When modeling, consider impacts to flow, trade-offs, costs, and alignment to SOT principles
First round of modeling
Testing your model

GET: How do we get information
• velocity
• WIP status

DELETE: How do we something from the team
• removal of backlog items

PUT: How do we introduce changes
• backlog order
• story elaboration

POST: How do we add something new
• backlog item
Second round of modeling
Catherine Louis
cll@cll-group.com
@catherinelouis

Raj Mudhar
rmudhar@deloitte.ca
@rmudhar

WHOLE-TEAM DYNAMIC ORGANIZATIONAL MODELING
A better way to design your organization

wtdom.org