Organizational Patterns in Software Development and Beyond

Valentino Vranić

Institute of Informatics, Information Systems, and Software Engineering

vranic@stuba.sk  fiit.sk/~vranic

11/12/2018
The principles on which PRINCE2 is based originate from lessons learned from projects both good and bad. They provide a framework of good practice for those people involved in a project. If a project does not adhere to these principles, it is not being managed using PRINCE2, because the principles are the basis of what defines a PRINCE2 project. The seven PRINCE2 principles can be summarized as:

- Continued business justification
- Learn from experience
- Defined roles and responsibilities
- Manage by stages
- Manage by exception
- Focus on products
- Tailor to suit the project environment.

People are crucial to the success of a project. It is not enough to have the required processes and systems in place: if the people on a project do not work effectively together, then the chances of the project’s success are severely restricted. Knowledge of different types of personalities and how these work together can help the Project Manager to structure balanced teams that can work together effectively during a project.

Different people have different characteristics, and certain types of people are more suited to certain roles. In a given environment, some combinations of personality types work better than others.
The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. **Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team.** Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity. The Scrum Team has proven itself to be increasingly effective for all the earlier stated uses, and any complex work.

Developer Controls Process

People don't like being ordered what to do, but the work needs to be organized.

Make the developers as a team decide how to organize development.

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies
contradicting forces

Developer Controls Process

People don't like being ordered what to do, but the work needs to be organized.

Make the developers as a team decide how to organize development.
```
<<interface>>
Subject

attach(observer: Observer)
detach(observer: Observer)
notify()

<<interface>>
Observer

update()

Subject1

state
+getState()
+setState()

Observer1

state
+update()
```
Observer

observing objects should be notified of the change in the state of the subject of their observation, and they should be attachable to the subject without having to modify it.
The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity. The Scrum Team has proven itself to be increasingly effective for all the earlier stated uses, and any complex work.

team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity. The Scrum Team has proven itself to be increasingly effective for all the earlier stated uses, and any complex work.


**Architect Also Implements**

Architects need to focus on the overall structure, but they should not loose contact with the development reality.

Let the (software) architect participate in actual programming.
Community Of Trust

People are naturally cautious and suspicious, which is often enforced by rules and practices, but for to really do the work, they need to trust each other.

Those "in charge" should make obvious they trust others by giving up the watch-over activities and letting people decide about their own work. Good and sincere communication is essential to overcoming fear.
The Water Cooler

People need to focus on the work within their projects to get it done, but to be creative, people need to be relaxed and informal, exchanging ideas over the project borders.

Encourage social structures that are unrelated to workplace structures and which will likely cut across the formal partitioning of the organization.
Google's fancy workspace is a giant water cooler pattern application.

Any problems that might come out of this?
Cone of Silence
Around
Osmotic Communication

PATTERN COMPOSITION
That's how an organization can grow

**Architect Controls Product** – establishes an architect role

**Architect Also Implements** – elaborates on that role making it also implement (develop)

**Developing In Pairs** – further precises how the architect collaboration with developers may be realized
Christopher Alexander

The Timeless Way of Building

A Pattern Language
Christopher Alexander

*The Timeless Way of Building*

*A Pattern Language*

Context

Independent Regions

... house cluster

House for a Small Family

Alcoves

...
How to learn organizational patterns?
Animate organizational patterns as text adventure games using Ericksonian conversational hypnosis language for an immersive experience

https://sites.google.com/a/scrumlopop.org/published-patterns/Organizational-Patterns-of-Agile-Software-Development/bookoutline/thepatternlanguages/organizationconstructionpatterns/peopleandcodepatternlanguage/architectalsoimplements

http://fiit.sk/~vranic/pub/ArchitectAlsoImplements/
Patterns come in pattern languages
That's how an organization can grow

**Architect Controls Product** – establishes an architect role

**Architect Also Implements** – elaborates on that role making it also implement (develop)

**Developing In Pairs** – further precises how the architect collaboration with developers may be realized
Patterns come in pattern languages

Represent a whole pattern language as yet another pattern
Good agile organizations exhibit many organizational patterns

Assess the agility of organizations according to the organizational patterns they apply and recommend further ones
Osmotic Communication in distributed software development?

➡️ Peripheral perception

Verbal stimuli: chat

https://github.com/ThePham/Indikom/

Visual stimuli
The idea of patterns come from building architecture, but the most lively experience with patterns is in software development.

Can this be transferred to other areas? Organizational of creative work, including teaching or coaching.

→ Drama patterns
Conway's Law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

How organizational patterns correspond to particular programming paradigms?

Organizational Patterns in Aspect-Oriented Software Development
Where do patterns come from?

Don't we bear them inside of ourselves?
Heart of Agile – Alistair Cockburn

Deliver
Collaborate
Reflect
Improve

http://heartofagile.com/
Collaborate

Learning

Income

Deliver

Insights

Reflect

Improve

Trust

Experiment

Change

Improvements
Scrum
Ivar Jacobson: Kill All Methods – Free the Practices
> Understanding and applying organizational patterns of software development

> Extrapolating the idea of patterns to other areas

> Otherwise, just be agile

[Link to online pattern library] tinyurl.com/patterns-zcu