BASTA!

Oliver Sturm

Title //
Presentation Name

Oliver Sturm (@olivers)

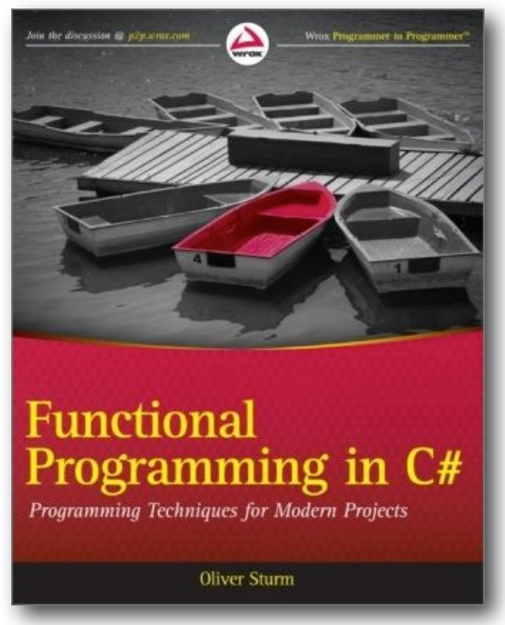
- **Consultant and Trainer, Author**
- **Associate Consultant at thinktecture**

- **.NET Application System Architecture**
  - User Interfaces
  - Data Handling / Data Access Architectures
  - Programming Languages
  - DevExpress Component/Framework Products

- **Microsoft MVP for C#**
- **INETA Europe Speaker**

- **Services: http://www.oliversturm.com**
- **Blog: http://www.sturmnet.org/blog**

- **oliver@oliversturm.com**



What I want to do today

- Talk about software architecture and the role it plays in the process of software creation
- Increase understanding between architects and developers
- Check out practical example scenarios

My own background

- Creating professional software since 1989
- Roles: Developer, Team Leader, Architect, Program Manager, Evangelist, CTO, Trainer, Consultant, ...
- Projects: Server, Client, client/server, UI, command line, Web, multi-tier, service-oriented, ...

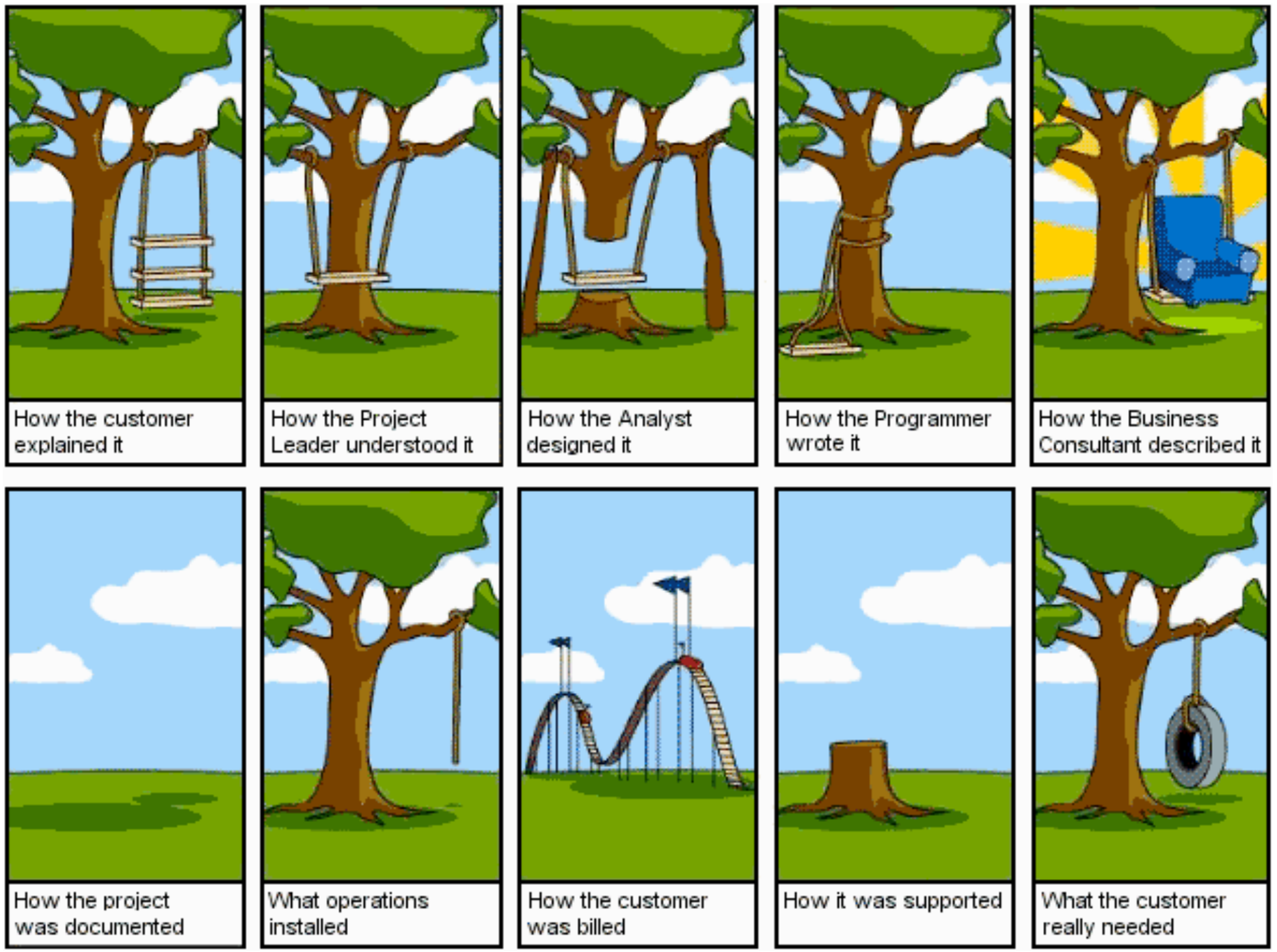
...

… but it'll only work with your help!

- No project is exactly like any other
- No two teams work exactly the same way
- There is no silver bullet for software

No silver bullet...

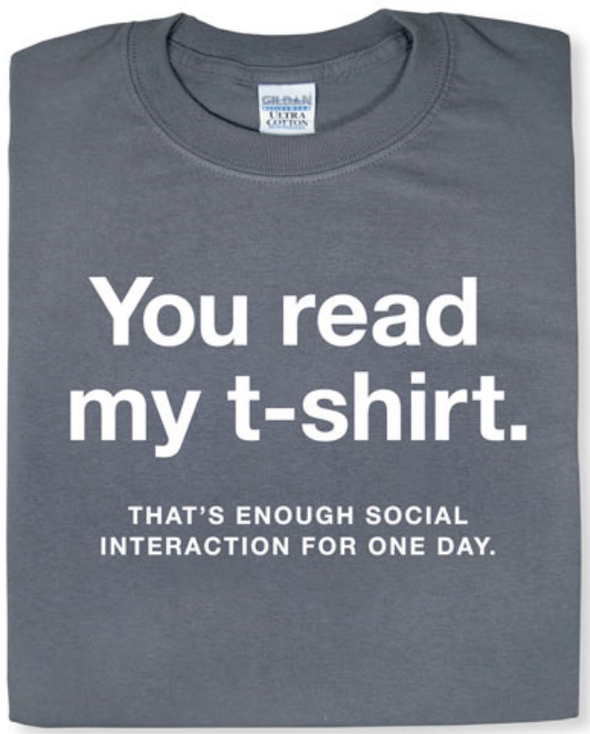
It's all about perspective...



Getting to know you - what do you do?

- Developer (...)
- Software Architect (...)
- ?

What do you think of... the other side?



What are the roles in software creation?

- Developer?
- Software Architect?
- ?

What are the steps through a project?

Project schedule

Deployment

(Customer?) Requirements

Testing

Agile process

Bug fixing

Architectural planning

Assigning tasks to developers

Documenting

Customer feedback

Software design

Programming

Assigning tasks to teams

Support

Architectural planning

do {
- What are the components?
- What happens inside each component?
- Which platform(s) to target?
- Which tools, languages, etc?
- How do components interact?

} while (!perfect);

How far should you go? Vision vs Agility

- Planning ahead can help find conceptional issues
- ... but it can also restrict you
- Agility is more about interpreting the importance of plans and their details correctly, not so much about not having any plans.

Correct interface definitions are important!

How do components interact? - Example

- Locally
  - Direct calls
  - RPC
  - Exchanging files
  - Message Queue
  - Services
  - Unifying platforms like WCF
  - ?

How do components interact? - Example

- Remotely
  - RPC
  - Services
  - Unifying platforms
  - ?
- Who should make this decision?
- Multi-tier? Client/server? Is that important? When do you decide it?

What influences decisions?

- Requirements?
- Management?
- Hype?
- Logic?
- Team abilities?
- ?

Let's build something slightly more complex

Example scenario

- Client: Cool Foods, manufacturer and seller of food products
- Recently created department “Cool Foods @Home”, home delivery
- Operating in 6 European countries
- Hierarchy:
  - Cool Foods Head Office UK
  - Cool Foods @Home Head Office Germany
  - Country Head Offices (6)
  - Delivery Offices (~30 per country), some of them franchisees

Example scenario - Requirements delivery offices

- Delivery vans (~20 per delivery office)
  - Bar code scanners - these have their own OS with APIs that can read product lists and similar information from databases (proprietary formats). Van drivers can use the scanners to register sales and take stock
  - Rugged tablets should be used to
    - provide drivers with routing information
    - communicate with the bar code scanners
    - feed back status info (sales/gps/...) to delivery offices
    - Provide access to customer management functionality like "register new", "adjust orders/ frequencies" and similar

Example scenario - Requirements delivery offices

- Receive products, pricing info, campaigns etc. from country head office
- Manually manage products, pricing info and campaigns (for franchisees)
- Manage food deliveries and stock levels
- Manage customers, orders, ordering subscriptions
- Manage personnel
- Interfacing with financial software packages (for franchisees)
- Flexible reporting
- Plan delivery routes
- Publish information to driver tablets and bar code scanners
- Receive status info from driver tablets
- Send sales, stock levels, personnel and financial information back to country home office
- Fleet management (for franchisees)

Example scenario - Requirements country head offices

- Manage products, pricing info and campaigns
- Manage suppliers and bulk ordering
- Publish information to delivery offices
- Receive information from delivery offices
- Flexible reporting
- Fleet management
- Interfacing with financial software packages
- Manage personnel
- Send accumulated stats to Head Office

Example scenario - Requirements Cool Foods @Home Head Office

- Manage products, pricing info and campaigns
- Manage suppliers and bulk ordering
- Publish information to country head offices
- Receive information from country head offices
- Flexible reporting
- Manage personnel
- Interfacing with SAP at Cool Foods General Head Office

What do we need to work out?

- What are the components?
- How do they interact?
- For each component:
  - What does it do?
  - What platform does it run on?
  - Which tools, languages and libraries are we going to use to write it?
  - Any more details you could/should specify here? Patterns?
- Common considerations:
  - Testing
  - Data access
  - Communication
  - Security
  - Deployment

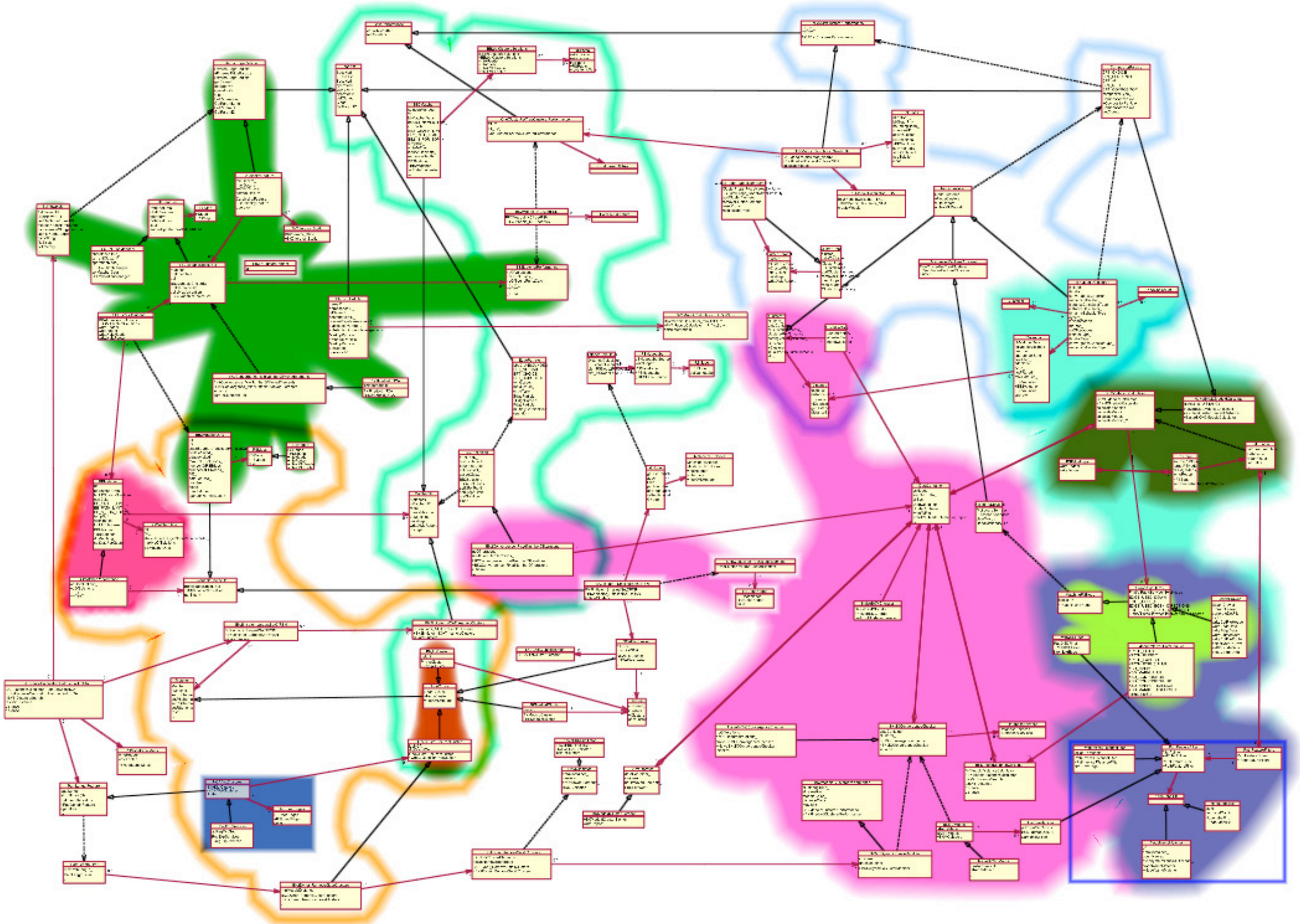
General notes

Component notes: ...

Even the simplest things...

A few thoughts on what comes next...

- Documentation
  - Do you like diagrams?
  - Written docs? Does anybody read them?
  - How do you make sure you're being understood?
- Communication
  1. Get the devs on board!
  2. Get the devs on board!
  3. Get the... oh, okay



Summary

- Architects and developers both have important roles to fill - whatever you want to call them!
- I hope the practical examples have turned out useful as well.

Thank you!

Please feel free to contact me about the content anytime!

oliver@oliversturm.com