THE PROBLEMS WE FACE

△ Process Manager
  ▸ Lots of interconnected data points
  ▸ Simulation of graphs / auto-detection of problems

△ Process Intelligence
  ▸ Auto-discovery of processes from data
  ▸ Problem detection using machine learning

△ Workflow Accelerator
  ▸ An IDE for a graphical programming language that does not suck
  ▸ Working with lots of data you don’t know
  ▸ Everything in a UI that Staplerfahrer Klaus needs to understand
INTRODUCTION

WHAT IS LEGACY?

- Dead code (code that isn’t used in your application anymore)
- “Finished” code: features that change so little that they are not being re-factored in a long time
- Too complex code that is very hard to change and therefore isn’t changed
- Good, re-factored code that needs to change because you changed your coding style or changed something in the technology stack.
NOTHING IS EVER PERFECT
YOU WILL NEED TO REFACTOR
YOU WILL NEED TO REFACTOR

WHAT THIS TALK IS ACTUALLY ABOUT

- Big technology changes
- Changes that take a long time to finish
- How not to piss off everyone else
WHAT HAPPENED?

THIS IS HOW YOUR APPLICATION LOOKS LIKE
WHAT HAPPENED?

THIS IS HOW YOUR APPLICATION LOOKS LIKE
WHAT HAPPENED?

HOW NOT TO DO IT

Signavio Workflow

Workflows

Actions

User Task

Assignment

Form

Access
WHAT HAPPENED?

HOW NOT TO DO IT

- Signavio Workflow
- Workflows
- Actions
- User Task
- Assignment
- Form
- Access

needs changes in
WHAT HAPPENED?

HOW NOT TO DO IT

Signavio Workflow

Workflows

Actions

User Task

Assignment

Form

Access

needs changes in

needs changes in
THIS IS A PROBLEM WHY?
THIS IS A PROBLEM WHY?

THIS IS HOW YOUR APPLICATION LOOKS LIKE

Diagram showing the structure of a workflow with nodes labeled as follows:
- Signavio Workflow
- Workflows
- Tasks
- Cases
- Analytics
- Trigger
- Actions
- Details
- Versions
- User Task
- Send Email
- Etc...
- Assignment
- Form
- Access
THIS IS A PROBLEM WHY?

THIS IS HOW YOUR APPLICATION LOOKS LIKE
THIS IS A PROBLEM WHY?

THIS IS HOW YOUR APPLICATION LOOKS LIKE
THIS IS A PROBLEM WHY?

THIS IS HOW YOUR APPLICATION LOOKS LIKE
THIS IS A PROBLEM WHY?

THIS IS HOW YOUR APPLICATION LOOKS LIKE

- Signavio Workflow
  - Tasks
  - Cases
  - Workflows
    - Trigger
    - Actions
      - User Task
        - Assignment
        - Form
        - Access
        - Send Email
  - Analytics
  - Details
  - Versions

FEATURE!
PEOPLE WILL HATE YOU
THE PROBLEM

WHY OTHER DEVELOPERS HATE YOU

- Your big refactoring adds tons of conflicts
- They need to fix those
- What they have built does not work anymore because of you
- They now work for you
THE PROBLEM

WHY PRODUCT MANAGEMENT HATES YOU

- You keep everyone busy but with no obvious result
- New features are not being added
- Existing features break
- Planning is impossible
THE PROBLEM

WHY SUPPORT HATES YOU

- Every time you ship, you destroy what was already there
- Customers get frustrated because things that used to work break and new features aren’t delivered to compensate for that
NO MORE FUN
FOR YOU
WHAT HAPPENED?
YOUR ASSUMPTIONS ARE WRONG

- The effects of your change aren’t that local at all
- The deeper you get into the rabbit hole the darker it gets
- What you thought would be a small easy change blows up into your face
- Since you block everything at top level you can only merge back to master when you’re done
THIS IS A PROBLEM WHY?

YOU SCREWED UP, WHAT NOW?

- Stop
- Think
- Revert
- Think
- Start again
OK, THEN HOW?
OK, THEN HOW?

A NEW HOPE

- Signavio Workflow
  - Workflows
    - Actions
      - User Task
        - Assignment
        - Form
        - Access
OK, THEN HOW?

A NEW HOPE
A NEW HOPE
OK, THEN HOW?

A NEW HOPE
THIS IS BETTER
WHY?
ITERATIVE ROCKS!
THIS IS HOW YOUR APPLICATION LOOKS LIKE
THIS IS HOW YOUR APPLICATION LOOKS LIKE
THIS IS HOW YOUR APPLICATION LOOKS LIKE
This is how your application looks like
THIS IS HOW YOUR APPLICATION LOOKS LIKE
WORK ITERATIVELY, BE READY FOR THE NEXT CHANGE

- Every change is isolated and has minimal effect
- Changes on other nodes along the way do not affect you
- You can merge your changes back to master after every node
- If unforeseen things happen, their impact stays local to what you are currently doing
RECAP
RECAP

SMALL ITERATIVE CHANGE OVER BIG, BLOCKING REFACTORIZING

Diagram showing a workflow with Signavio Workflow, Workflows, Actions, User Task, Assignment, Form, Access. The workflow is represented with a green line indicating changes are possible. The diagram on the right shows a similar structure but with red lines indicating changes are needed.
FEEDBACK?
FURTHER READING

- https://www.youtube.com/watch?v=BF58ZJ1ZQxY
- https://mikadomethod.wordpress.com/