



**ČVUT**





ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE

# **Project CASA: Final Project Presentation**

**May 25, 2017**

**Yevgeniya Chekh, Jan Kohout, David Löffler, Miroslav Rudišin, Kryštof Sýkora, Marek Szeles, Ho Minh Thanh**

# Presentation agenda

-  **Introduction, what is CASA?**
-  **Review formal project outputs**
-  **Project run overview**
-  **Lessons learned, individual reviews**

# Who are we?

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# What is CASA?

***Dr. Bestoun S. Ahmed →***



Speciální projekt v angličtině - bonus 10 bodů navíc

Rewriting of Casa combinational testing tool to Java

Zanalyzujte open source aplikaci CASA napsanou v C++ a přepiště ji do jazyka Java nebo C# (.NET). Zdrojový kód je k dispozici. Výsledný program by měl fungovat stejně rychle jako zdrojový. Součástí projektu bude provést srovnání rychlosti.

# What is CASA?

***Dr. Bestoun S. Ahmed →***



## 1. Project information

Project CASA is part of a larger ongoing academic project, information on it can be found here:  
[Quality Assurance and testing methodology for IoT](#)

Since the main deliverable of the project is having working code at the end, we have opted for agile project management, which means less strict rotation of project roles with more iterations between shorter development phases.

The main use case is to recreate the original CASA program 1 to 1 in relation to structure. Some parts of the program, namely the “sat” and “minisat” packages might be online in Java already, which will save time.

**So... Is CASA a success?**

**Yes!**

**No!**



**Jein?!**

**Why?**

**Meetings: 16**

**Steering Workshops held: 5 (so far)**

**3 Presentations (excluding this one)**

**Documents created: 28**

**Spanning over: 118 pages**

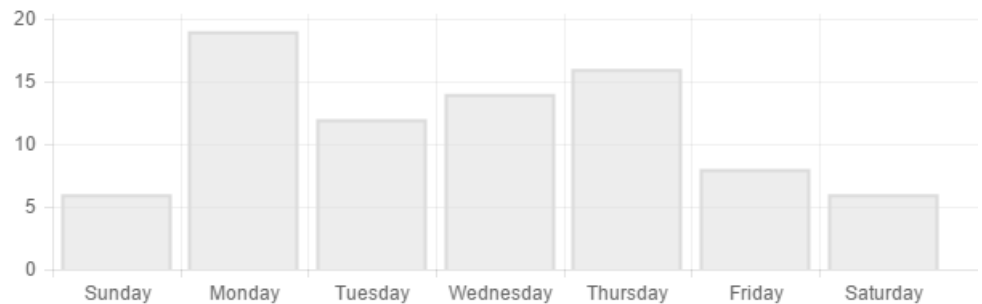
**Lines of code worked on: 323 198**

## Also why?

Commit statistics for **master** Feb 24 - May 25

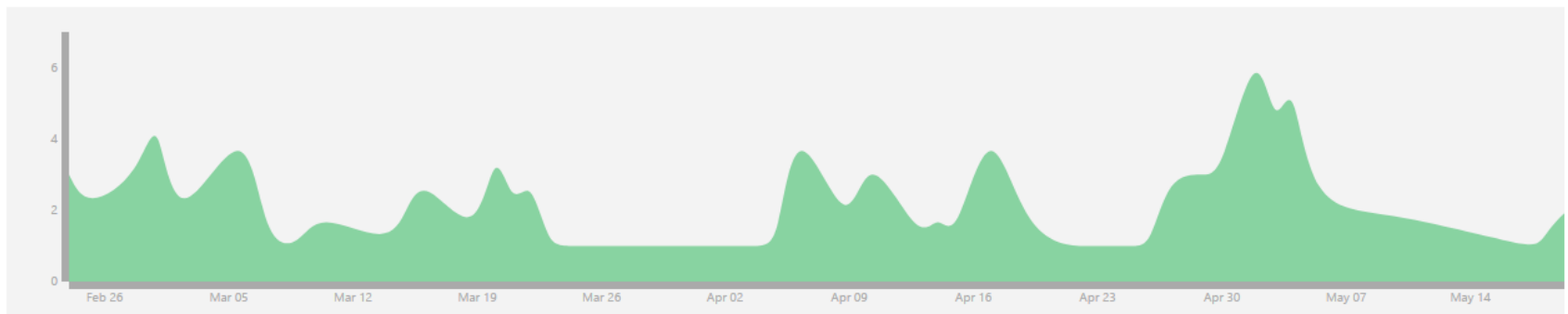
- **113** commits during **90** days
- Average **1** commits per day
- Contributed by **8** authors

Commits per weekday



February 24 2017 - May 25 2017

Commits to master, excluding merge commits. Limited to 6,000 commits.





## And finally...

**Project CASA: Acceptance document**

**B6B36RSP FEE CTU, Summer Semester 2017**

**Authors:** Yevgeniya Chekh, Jan Kohout, David Löffler, Kryštof Sýkora, Marek Szeles, Ho Minh Thanh, Miroslav Rudišin

# Project CASA: Acceptance document

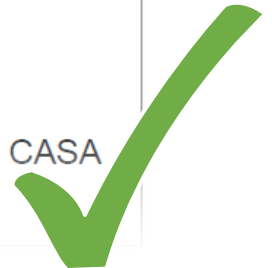
---

**Project team:** Yevgeniya Chekh, Jan Kohout, David Löffler, Kryštof Sýkora, Marek Szeles,  
Ho Minh Thanh, Miroslav Rudišin

**Client:** Ing. Miroslav Bureš, Ph.D.; Dr. Bestoun S. Ahmed Al-Beywanee, Ph.D.

## 1. Document description

This document serves as a record that all of the Project CASA's deliverables have been accepted as complete and states that all acceptance criteria have been met. The Project CASA can by this document be formally considered as closed.



## Let's take a look at the official rules though:

položka	počet bodů
1 Projektový plán	2
2 Aktualizovaný risk log	1
3 Katalog detailních požadavků	2
4 Specifikace případů užití	2
5 Dokumentace business procesů v systému	2
6 Model architektury	1
7 Deployment model	1
8 Prototyp GUI	2
9 Implementovaná aplikace	9
10 Testovací strategie	2
11 Testovací scénáře	1

## Let's take a look at the official rules though:

položka		počet bodů
1	Projektový plán	2 ✓
2	Aktualizovaný risk log	1 ✓
3	Katalog detailních požadavků	2 ✓
4	Specifikace případů užití	2 ✓
5	Dokumentace business procesů v systému	2 ✗
6	Model architektury	1 ✓
7	Deployment model	1 ✓
8	Prototyp GUI	2 ✗
9	Implementovaná aplikace	9 ✓
10	Testovací strategie	2 ✗
11	Testovací scénáře	1 ✗



# 1 Project plan

Project CASA: Project Plan B6B36RSP FEE CTU, Summer Semester 2017  
Authors: Yevgeniya Chekh, Jan Kohout, David Löffler, Kryštof Sýkora, Marek Szeles, Ho Minh Thanh, Miroslav Rudišin

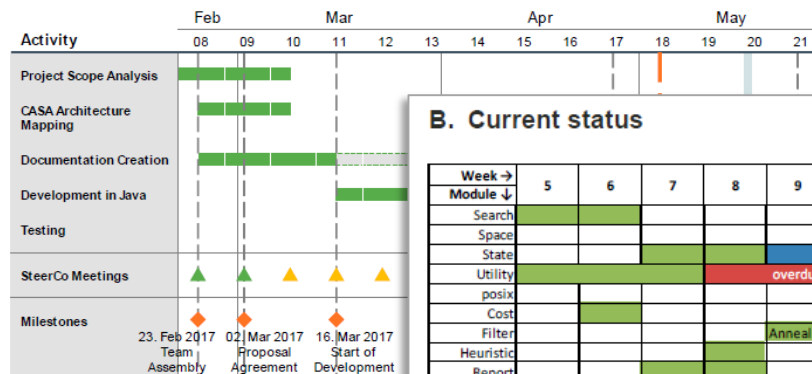
## Project CASA: Project Plan

Last update: May 3 2017

### Contents

1. Project information
2. Overall project
  - A. Baseline – Initial
  - B. Current status
3. Development plan
  - A. Baseline – Initial
  - B. Current status
4. Testing plan
  - A. Baseline – Initial
  - B. Current status

### B. Current status



### B. Current status

Week → Module ↓	5	6	7	8	9	10	11	12	13	14	Responsible
Search	Green	Green									Miroslav Rudišin
Space											David Löffler, Jan Kohout
State			Green	Green	Green						Miroslav Rudišin, Kryštof Sýkora, David Löffler
Utility	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Jan Kohout, David Löffler, Miroslav Rudišin
posix											David Löffler, Miroslav Rudišin
Cost		Green									David Löffler, Miroslav Rudišin
Filter					Green	Green	Green	Green	Green	Green	David Löffler
Heuristic					Green	Green	Green	Green	Green	Green	David Löffler
Report					Green	Green	Green	Green	Green	Green	Yevgeniya Chekh
Goal					Green	Green	Green	Green	Green	Green	David Löffler
IO											ChR by David Löffler
Bookkeeping											Yevgeniya Chekh, David Löffler
Events											David Löffler
sat/minisat											David Löffler
Annealing											David Löffler
Binary search											Jan Kohout



## 2 Risk log

	A	B	C	D	E	F	G	H	I	J
	ID	Risk event	Owner	Probability	Impact	Priority	State	If happened, when?	Risk area	Proposed solution
1	R1	Realization team change	CASA team	High	Low	B	Resolved	Week 2	Global	Redistribution of project work/Changing project scope
2	R2	Not being able to deliver full output	CASA team/Client	High	Medium	B	Resolved	Week 13	Global	Changing project scope
3	R3	Not being able to translate (parts of) program	CASA team	Medium	Medium	B	Resolved	Week 11	Global	Asking for external help
4	R4	Radical change of requirements	CASA team	Medium	High	C	Resolved	Week 6	Global	Refusing change request within bounds of project
5	R5	Overall project cancelled	Client	Low	High	B	Closed	-	Global	Change project scope, reflect in final output
6	R6	Development Team member lost	CASA Team	Low	Medium	B	Closed	-	Development	Change project scope, redistribute work
7	R7	Testing Team member lost	CASA Team	Low	Low	A	Closed	-	Testing	Change project scope, redistribute work
8	R8	Analysis Team member lost	CASA Team	Low	Low	A	Closed	-	Analysis	Change project scope, redistribute work
9										
10										
11										

R4	Radical change of requirements	Immediately	Project Leader	X	-	X	-	
R5	Overall project cancelled	Immediately	Project Leader	-	-	X	-	
R6	Development Team member lost	1 week after change		X	-	X	X	
R7	Testing Team member lost	1 week after change		X	-	X	X	
R8	Analysis Team member lost	1 week after change		X	-	X	X	



## 3 Detailed requirements specification

Project CASA: Requirements

B6B36RSP FEE CTU, Summer Semester 2017

Authors: Yevgeniya Chekh, Jan Kohout, David Löffler, Kryštof Sýkora, Marek Szeles, Ho Minh Thanh, Miroslav Rudišin

### Project CASA: Requirements

*Last update: May 3 2017*

#### 1. Requirements

##### A. Initial Requirements

These requirements were formulated at project start, after about three days of analyzing the general assignment.

1. Write new program in Java with same functionality as CASA.
  - 1.1. Create covering array with test cases created by pairwise testing method.
  - 1.2. Using constraints that affects calculation of covering array.
  - 1.3. Set the seed value for the random number generator.
  - 1.4. Set the initial number of iterations allowed at each array size.
  - 1.5. Set the number of retries allowed at the same array size.
  - 1.6. Set the weight of the upper bound in the binary search partition.
  - 1.7. Set the initial temperature.
  - 1.8. Set the temperature multiplier applied each iteration.
  - 1.9. Let the covering array be no smaller than the given size.
  - 1.10. Let the covering array be no larger than the given size.
  - 1.11. Lock the covering array at the given size.
2. Keep the CASA runtime speed in comparison to original in C++.

##### B. Revised Requirements

1. Write new program in Java with same functionality as CASA.
  - 1.1. Create covering array with test cases created by pairwise testing method.
  - 1.2. Using constraints that affects calculation of covering array.

##### C. Operational project requirements

1. Find SAT and minisat module in Java.
2. Find a way to use code in C language from Java.
3. Compare Choco solver with SAT solver in Java.



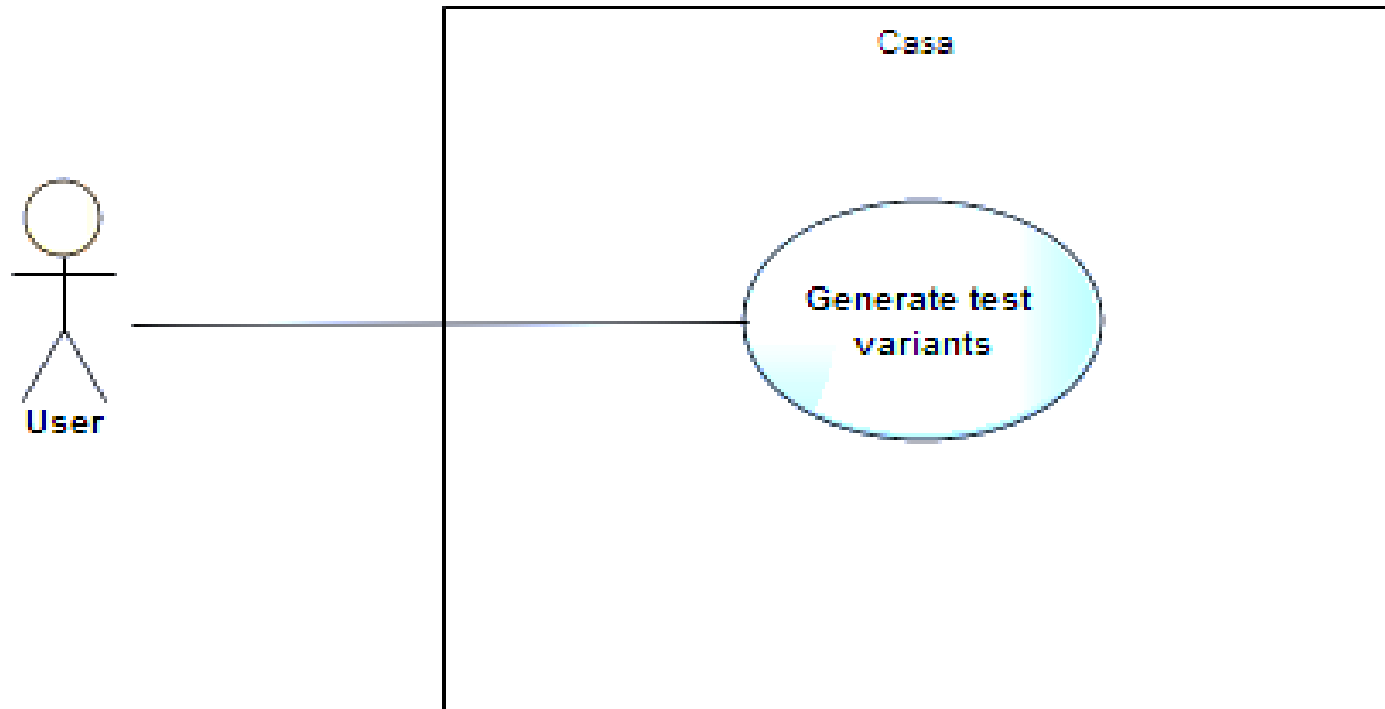
## 3 Bonus – Change requests

ID	Date	Minutes relevant	Minutes task ID	Description	Proposed by	Scope	Project Team Proposed action	Accepted?
ChR1	28.3.2017	W6	-		Client - Bestoun S. Ahmed	Global	NO	NO
ChR2	28.3.2017	W6	-	Program performance is not a priority issue anymore	Client - Bestoun S. Ahmed	Global	YES	YES
ChR3	6.4.2017	W5,W7	5.1	Possibility of running C++ code snippets from Java	CASA Team - Kryštof Sýkora	Development	YES	NO
ChR4	27.4.2017	W10	-	Simplifying parsing to exclude posix and IO modules	CASA Team - David Löffler	Development	YES	YES
ChR5	4.5.2017	W11	-	Project acceptance requirement change	CASA Team - Marek Szeles	Global	YES	YES
ChR6	9.5.2017	W12	-	Project scope can be extended with teammembers participating receiving compensation for finishing of implementation	Client - Miroslav Bureš	Global	YES	Pending

Proposed by	Scope	Project Team Proposed action	Accepted?	As proposed by project team?	Note
Client - Bestoun S. Ahmed	Global	NO	NO	YES	
Client - Bestoun S. Ahmed	Global	YES	YES	YES	
CASA Team - Kryštof Sýkora	Development	YES	NO	NO	Separate document analysis available
CASA Team - David Löffler	Development	YES	YES	YES	Separate document analysis available
CASA Team - Marek Szeles	Global	YES	YES	YES	
Client - Miroslav Bureš	Global	YES	Pending	NO	

✓ *Completed*

## 4 Use case specification (1/3)







## **4 Use case specification (2/3)**

### **Generate test cases**

The general use case of the program is to generate covering array of test cases. There are few parameters.

- (Required) Model file with
  - o strength of testing
  - o number of options
  - o number of values in each option
- (Optional) Constraint file with
  - o Number of disjunctive clauses
  - o Number of terms in each disjunctive clause
  - o Terms itself from each clause
- (Optional) Name of the Output file



## 4 Use case specification (3/3) – ChR4

### Project CASA – Change request: Parsing with IO/posix modules

Modules posix and IO are used for parsing the input file. Right now the original CASA program has several possible combinations of input.

```
" -o, --output      [FILE]  write to the given file\n" -c, --constrain    [FILE]  incorporate the given constraint file\n"\n"\n" -s, --seed        [SEED]  set the seed\n"\n" -i, --iterations  [COUNT] set the number of iterations\nsize\n" -r, --retries     [COUNT] set the number of retries\n" -p, --partition   [RATIO]  set the weight of the partition\npartition\n"\n"\n" -t, --temperature [TEMP]  set the temperature\n" -d, --multiplier  [RATIO]  set the multiplier\n"\n"\n" -l, --lower-bound [SIZE]  let the covering array be at least of the given size\n" -u, --upper-bound [SIZE]  let the covering array be at most of the given size\n" -n, --known-size  [SIZE]  lock the covering array at the given size\n"\n"\n" -v, --version      show the current version and exit\n" -h, --help         show this help and exit
```

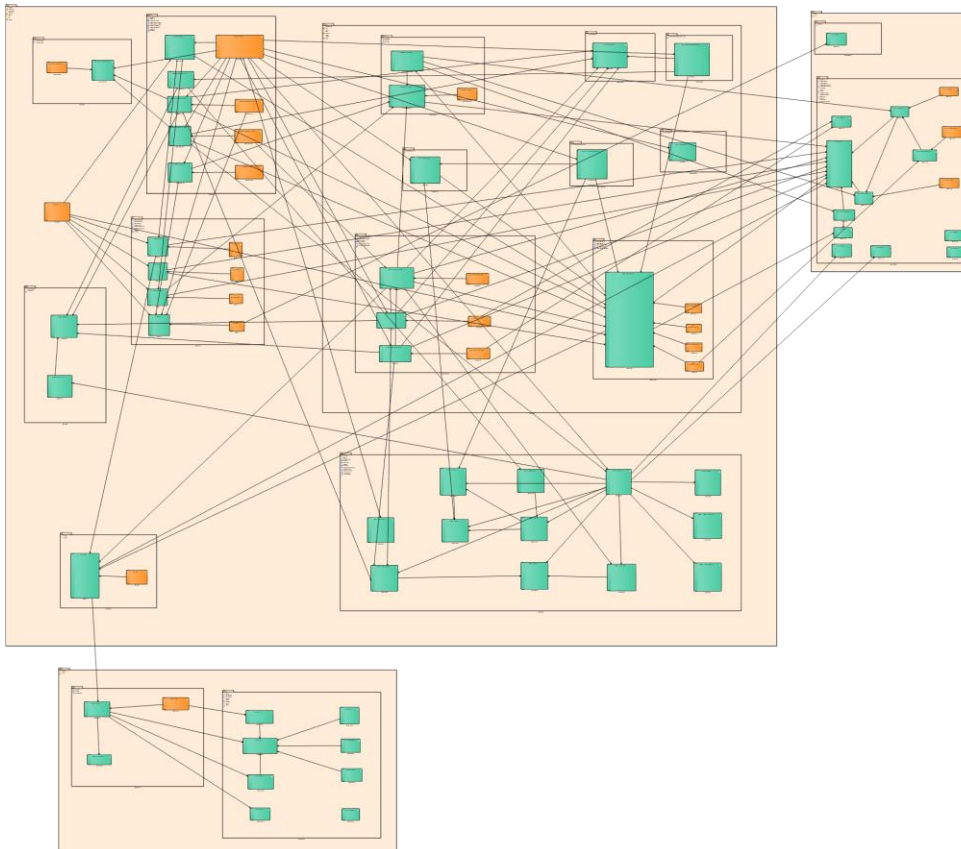
So far we do not require most of the combinations of input. Considering this fact and our limited time we have decided to not implement these posix and IO because we only need these following rows:

```
" -o, --output      [FILE]  write to the given file, regardless of seed\n" -c, --constrain    [FILE]  incorporate the given constraint file
```

This is the reason why we will implement our own input parser

✓ *Completed*

## 6 Architecture model v1



### Legend



Header file



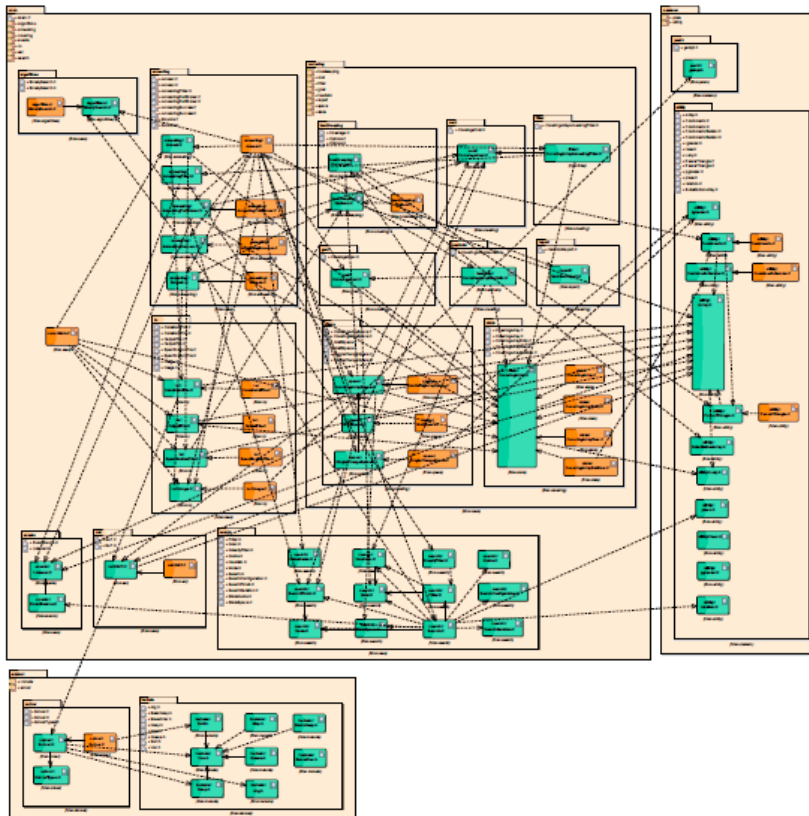
C file



Dependency

✓ *Completed*

## 6 Architecture model v2



### Legend



Header file



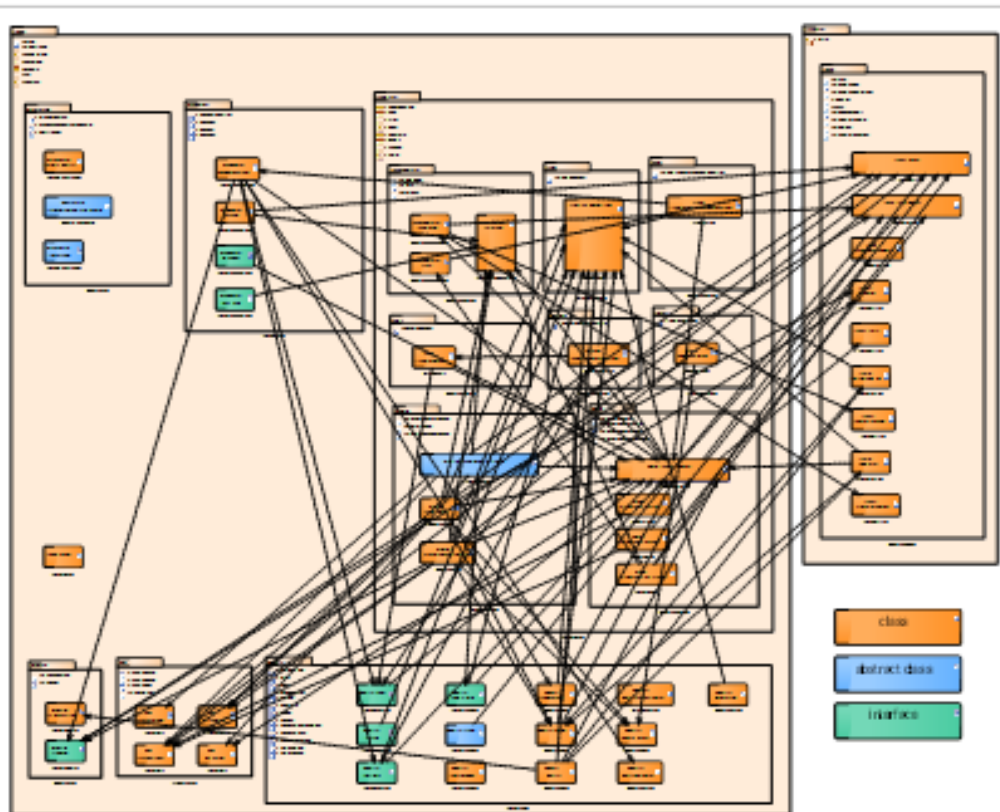
C file







Dependency

✓ *Completed*

## 6 Architecture model v3 – Java

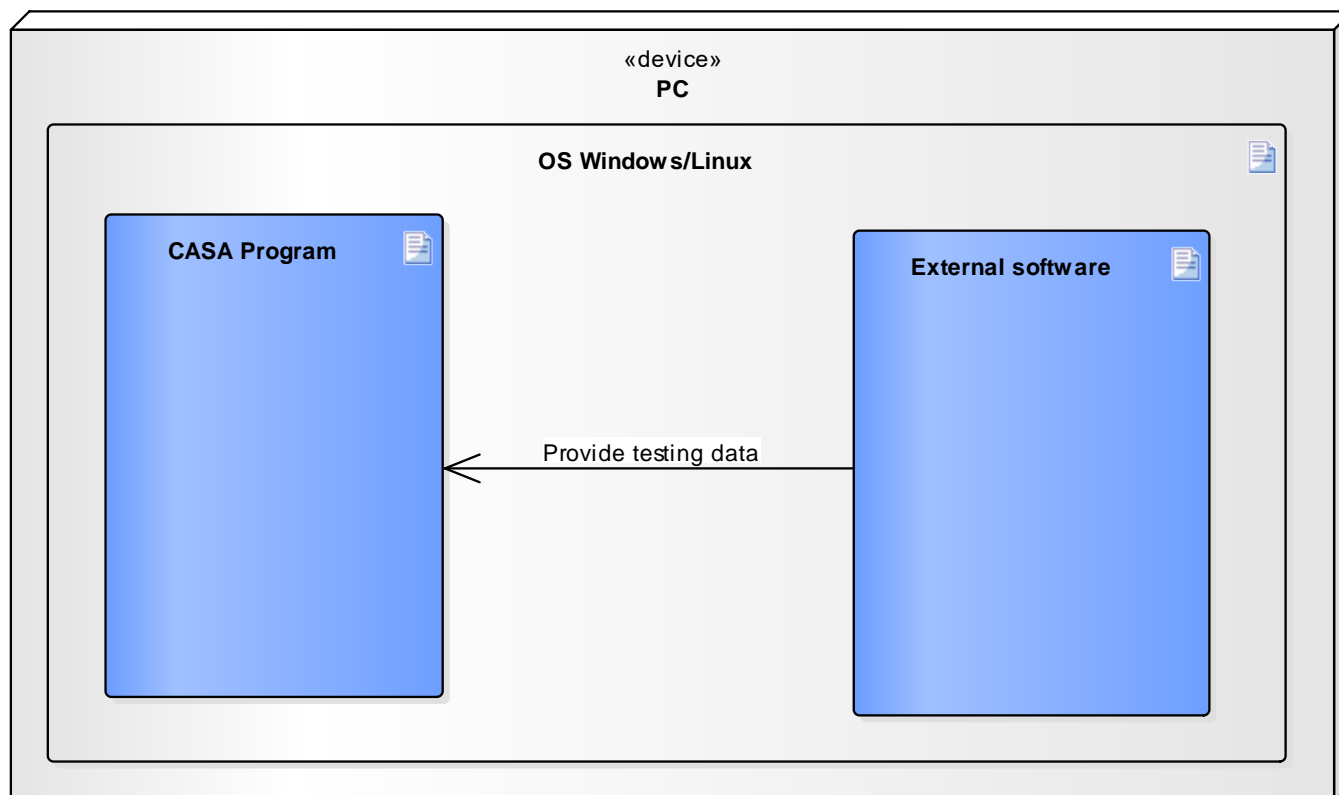


### Legend

-  Class
-  Abstract class
-  Interface
-  Dependency



## 7 Deployment model





## 9 Application – Acceptance + Change of scope



**Accepted current result,  
next steps TBD**

**X** *Out of scope*

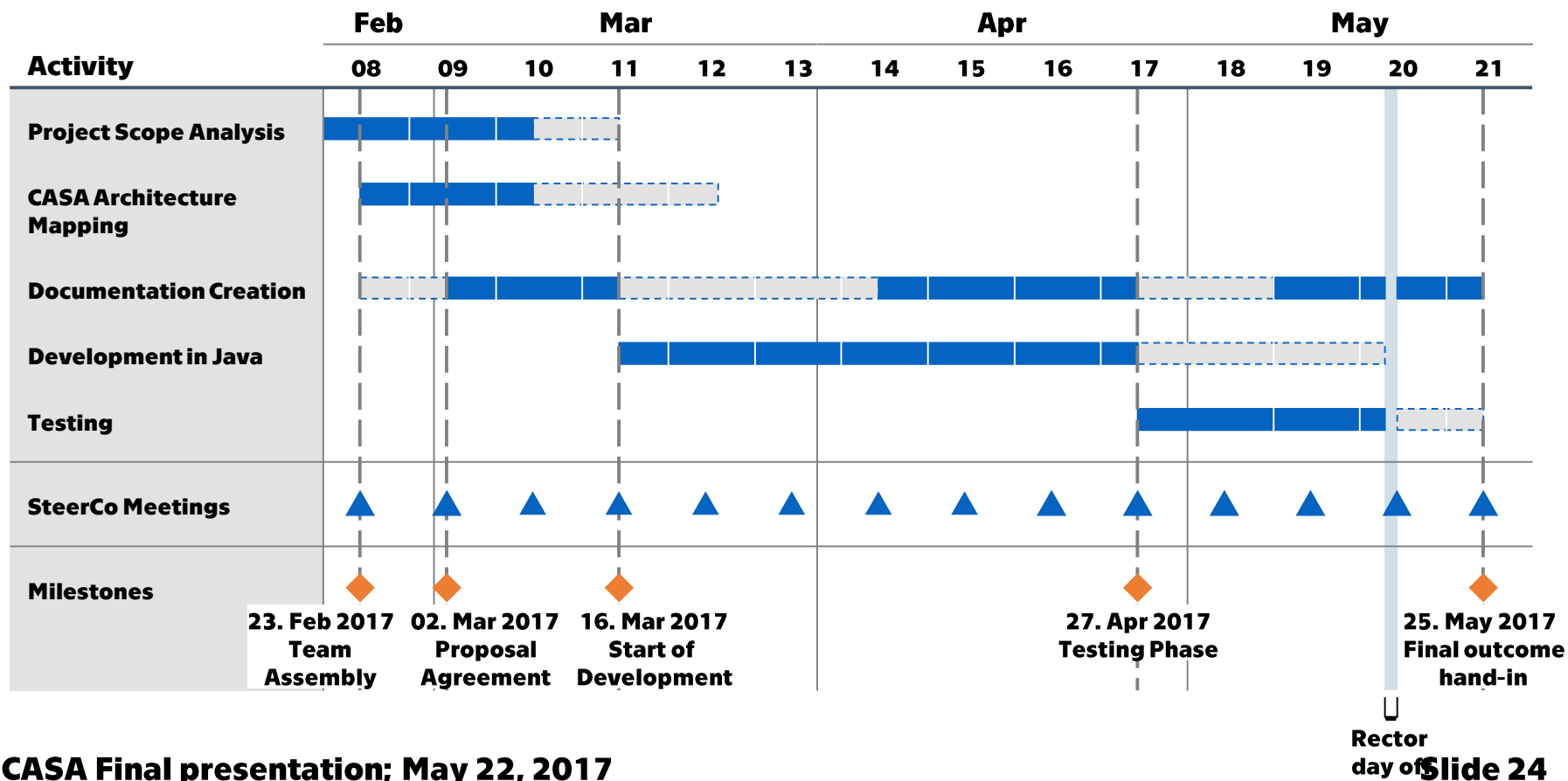
- 5 8 Business processes; GUI; Testing strategy:**  
**10 11 Out of project scope/other solution**



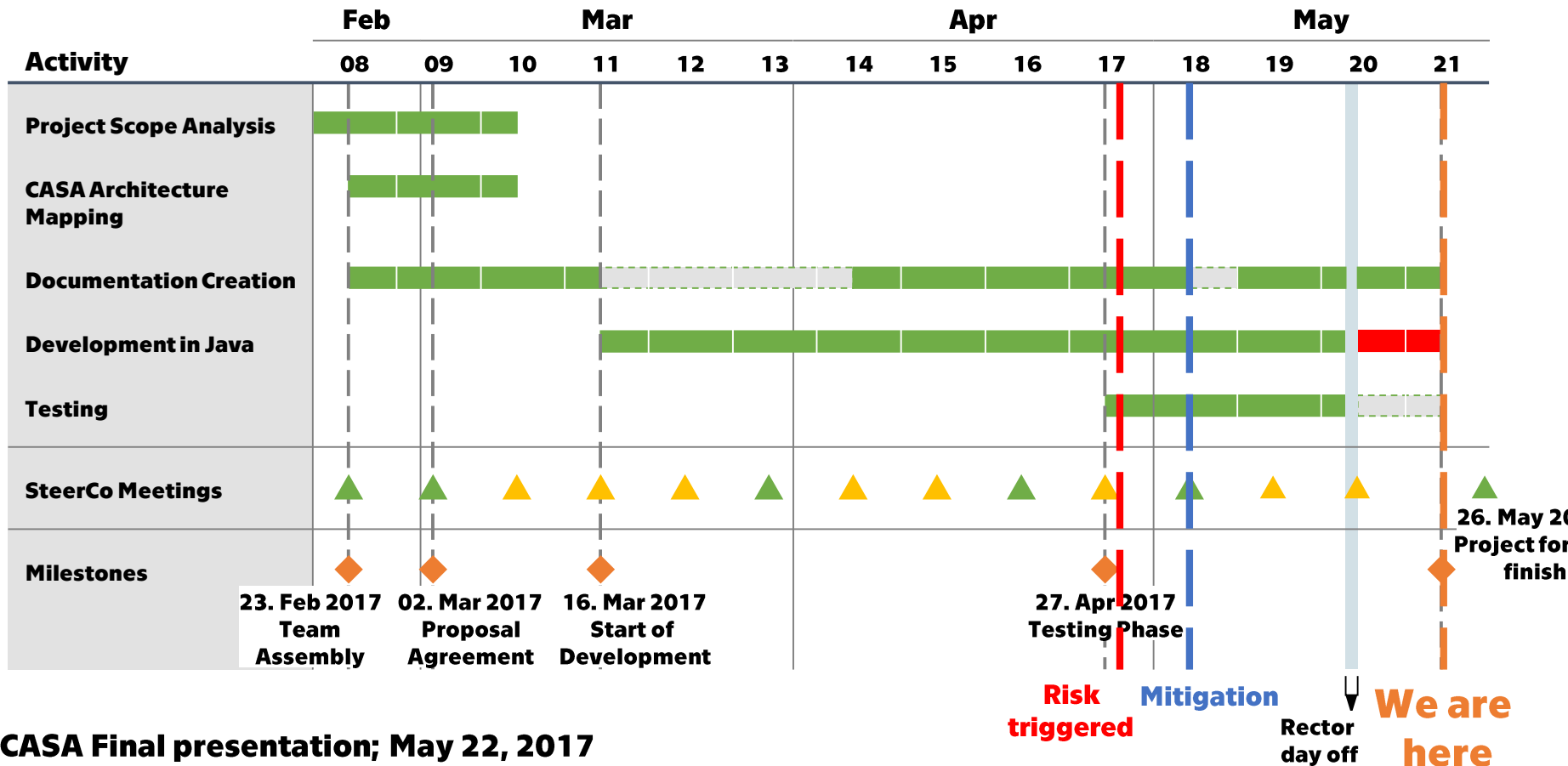
**Not relevant for project  
(accepted)**



# Project plan - expectation



# Project plan - reality



# Critical moment: Development stall

## >Risk management with project sponsor

### Project CASA: Help needed with C++ to Java translation

#### 1. Proposition

Project CASA is about translating C++ code into functional Java counterpart. The bulk of the project is completed, however some parts are beyond the abilities of the team members.

Thus, we are seeking someone with knowledge of both C++ and Java programming to successfully finish the project.

*Note: code contained here is only for reference, any potential co-worker would gain access to the project [GitHub repository](#).*

#### 2. Sample problems

##### 2.1. CoveringArray (1)

C++ CODE (CASA original):

```
void CoveringArray::setTrackingCoverage(bool trackingCoverage) {
    if (this->trackingCoverage) {
        this->trackingCoverage = trackingCoverage;
    }
}

bool CoveringArray::isTrackingCoverage() const {
    return trackingCoverage;
}

void CoveringArray::setStrength(int strength) {
    unsigned limit = coverage.getOptions().getFirst();
    Array<unsigned> firsts = coverage.getOptions().getFirstSymbols();
    Array<unsigned> counts = coverage.getOptions().getSymbolCounts();
    coverage.setStrength(strength);
    coverage.setOptions(0);
    coverage.setOptions(0);
    if (substitutions.getOptions().size() > 0) {
        unsigned hint = 0;
        for (Array<unsigned> column = combineAdicBegin(strength),
              subAdic(strength);
              column[strength - 1] < limit;
              combineAdicColumn(column), ++hint) {
            for (unsigned i = 0; i < firsts.size(); ++i) {
                for (unsigned j = 0; j < counts[i]; ++j) {
                    subAdic[i] = ("this" + column[j]);
                }
            }
        }
    }
}
```

Page 1 of 10

```
void CoveringArray::setTrackingCoverage(bool trackingCoverage) {
    if (this->trackingCoverage) {
        this->trackingCoverage = trackingCoverage;
    }
}

bool CoveringArray::isTrackingCoverage() const {
    return trackingCoverage;
}

void CoveringArray::setStrength(int strength) {
    unsigned limit = coverage.getOptions().getFirst();
    Array<unsigned> firsts = coverage.getOptions().getFirstSymbols();
    Array<unsigned> counts = coverage.getOptions().getSymbolCounts();
    coverage.setStrength(strength);
    coverage.setOptions(0);
    coverage.setOptions(0);
    if (substitutions.getOptions().size() > 0) {
        unsigned hint = 0;
        for (Array<unsigned> column = combineAdicBegin(strength),
              subAdic(strength);
              column[strength - 1] < limit;
              combineAdicColumn(column), ++hint) {
            for (unsigned i = 0; i < firsts.size(); ++i) {
                for (unsigned j = 0; j < counts[i]; ++j) {
                    subAdic[i] = ("this" + column[j]);
                }
            }
        }
    }
}
```

JAVA CODE (CASA - TRANSLATION):

```
public void setTrackingCoverage(boolean trackingCoverage) {
    if (this.trackingCoverage) {
        this.trackingCoverage = trackingCoverage;
    }
}

boolean isTrackingCoverage() {
    return trackingCoverage;
}

void setStrength(int strength) {
    Integer limit = coverage.getOptions().getFirst();
    Integer hint = coverage.getOptions().getFirst();

    Vector<Integer> firsts = coverage.getOptions().getFirstSymbols();
    Vector<Integer> counts = coverage.getOptions().getSymbolCounts();
    coverage.fill(0);
    coverage.setStrength(strength);
    coverage.setOptions(0);
    coverage.setOptions(0);
    if (substitutions.getOptions().size() > 0) {
        Integer hint = 0;
        for (Integer column = combineAdicBegin(strength);
              column[strength - 1] < limit;
              combineAdicColumn(column), ++hint) {
            for (Integer i = 0; i < firsts.size(); ++i) {
                for (Integer j = 0; j < counts[i]; ++j) {
                    subAdic[i] = ("this" + column[j]);
                }
            }
        }
    }
}
```

Page 2 of 10

# Risk management failed > Change of scope



**Project scope change**

# Deep Dive: Analytics

# **Analysis – What are we doing?**

**First step: Read the documentation - success**

**Second step: Understand the documentation – failed**

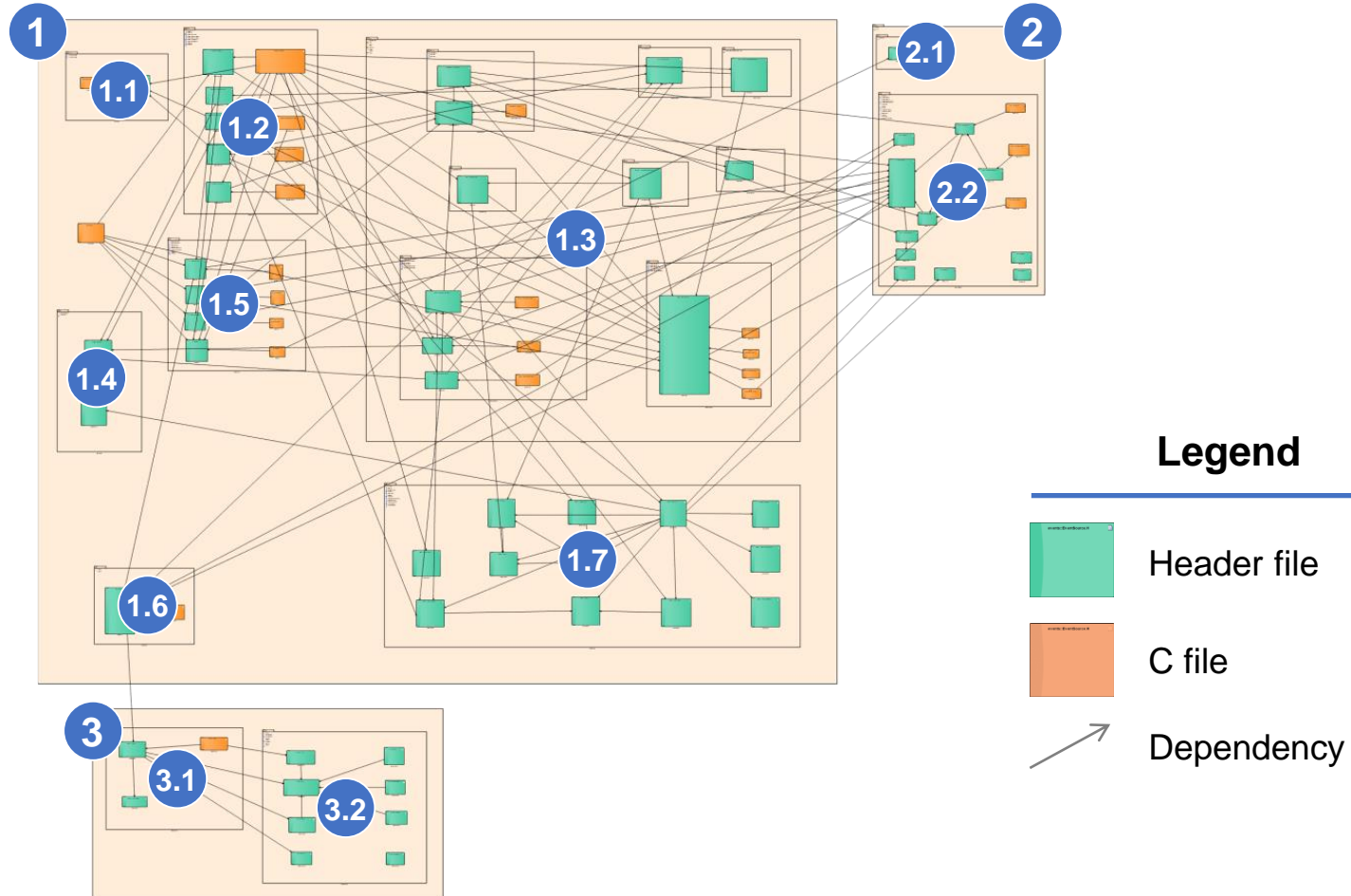
**Third step: Ask Bestoun S. Ahmed - success**



ČVUT

ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE

# Analysis - Full Project View



**Analysis – We don't want to reinvent wheels.**

**What modules can we run in C++?**

**– None**

**Are there modules in Java we can use?**

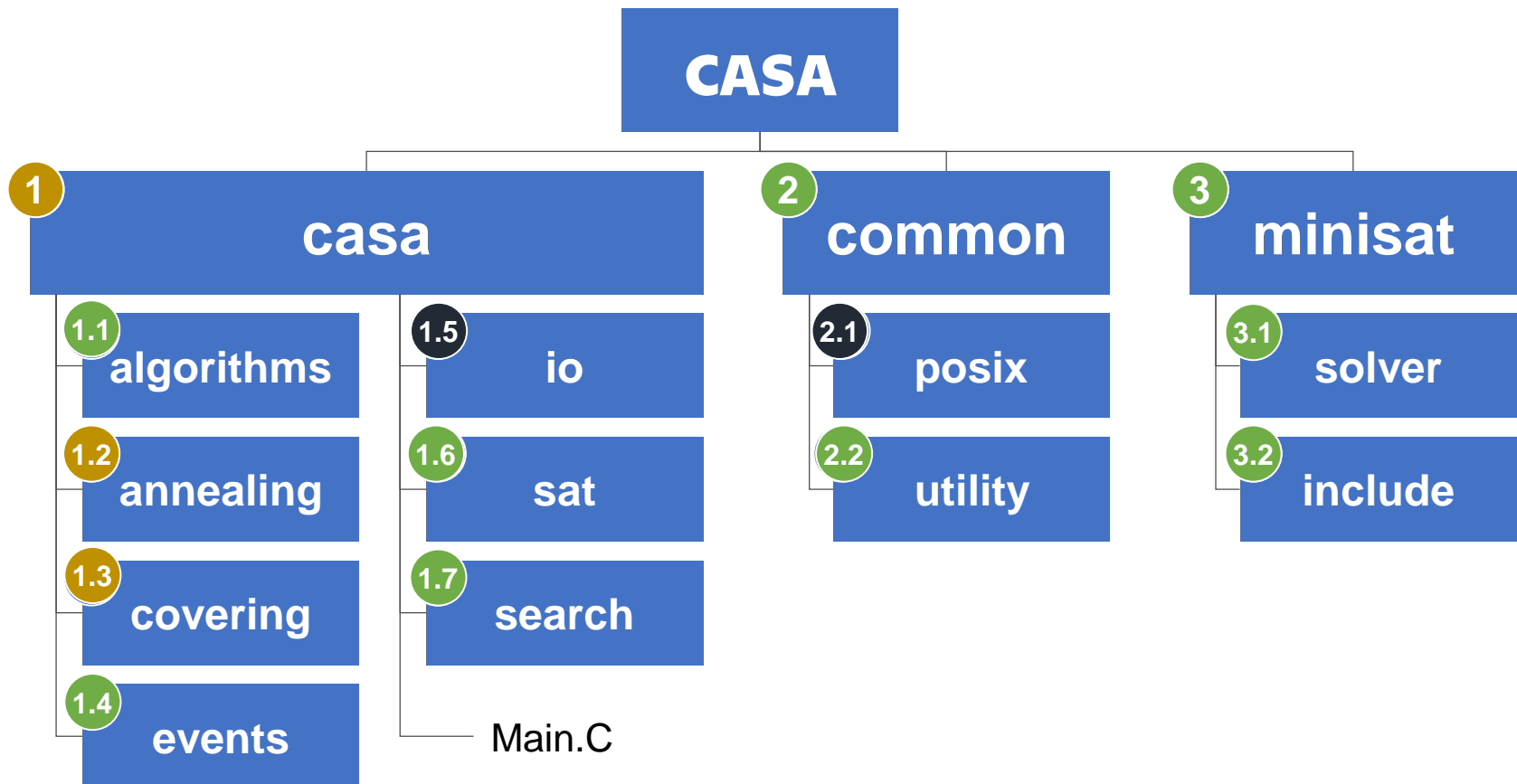
**– Yes, SAT**



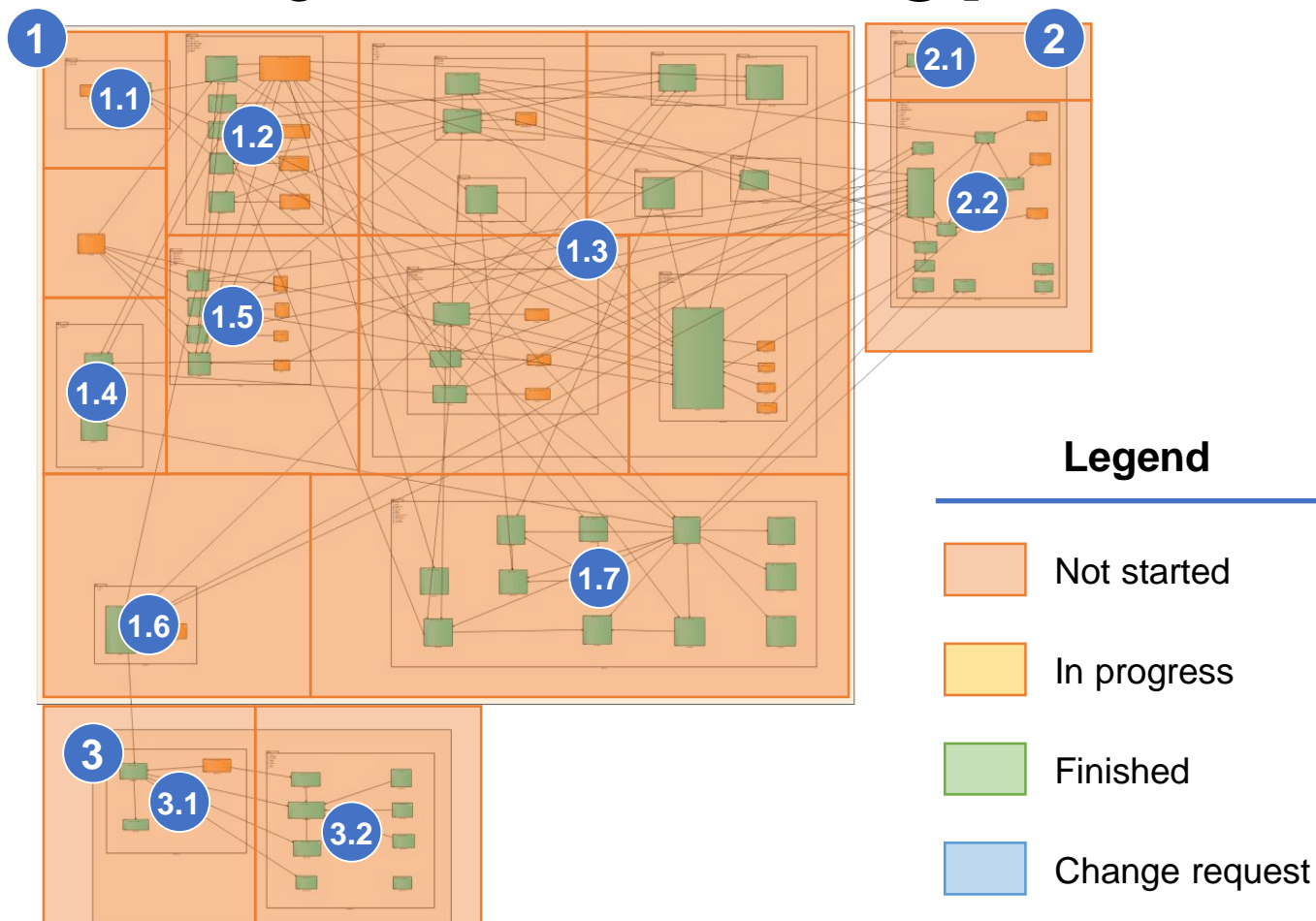
# Deep Dive: Development

# Development progress

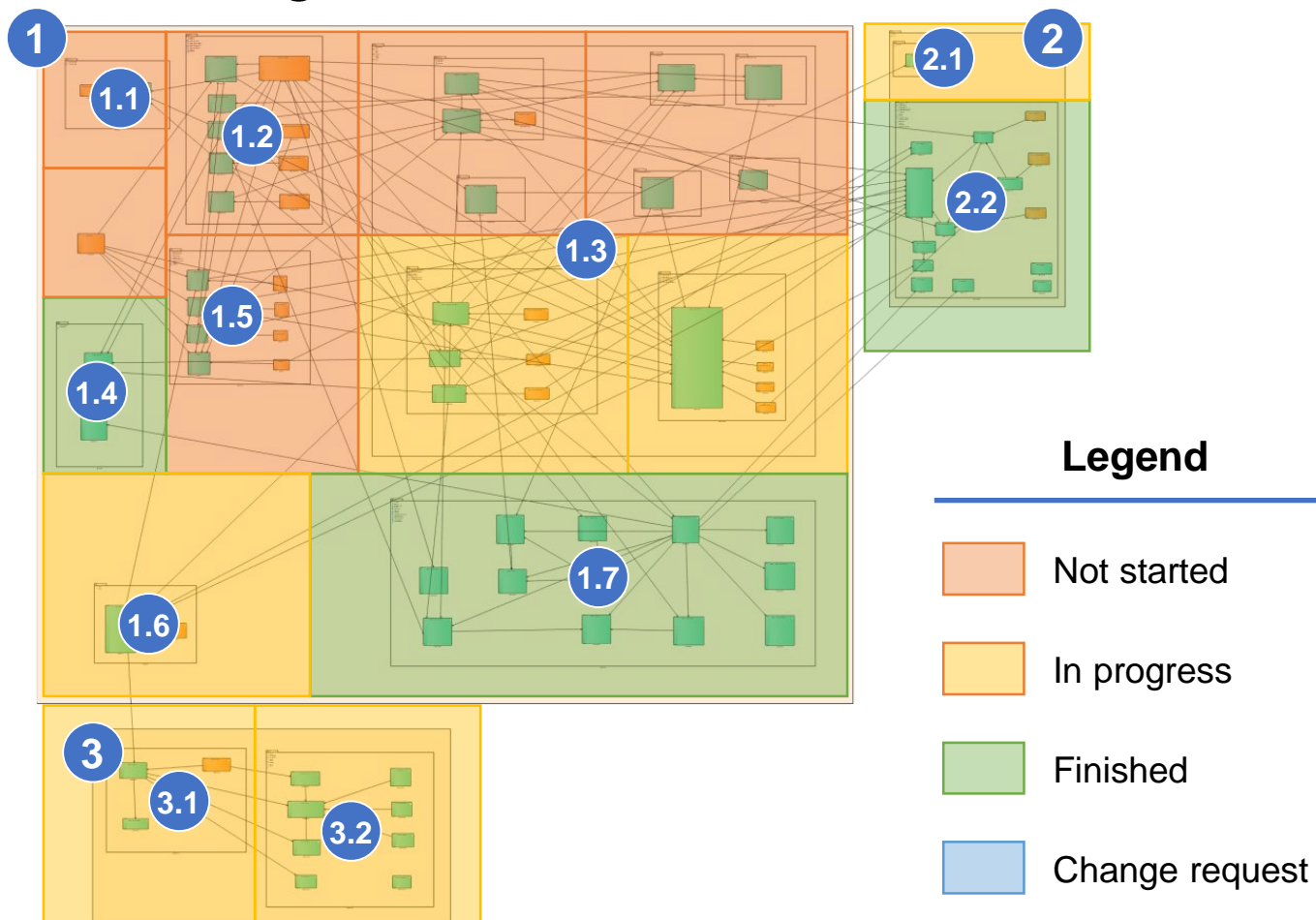
- **Starting with package with least dependencies**



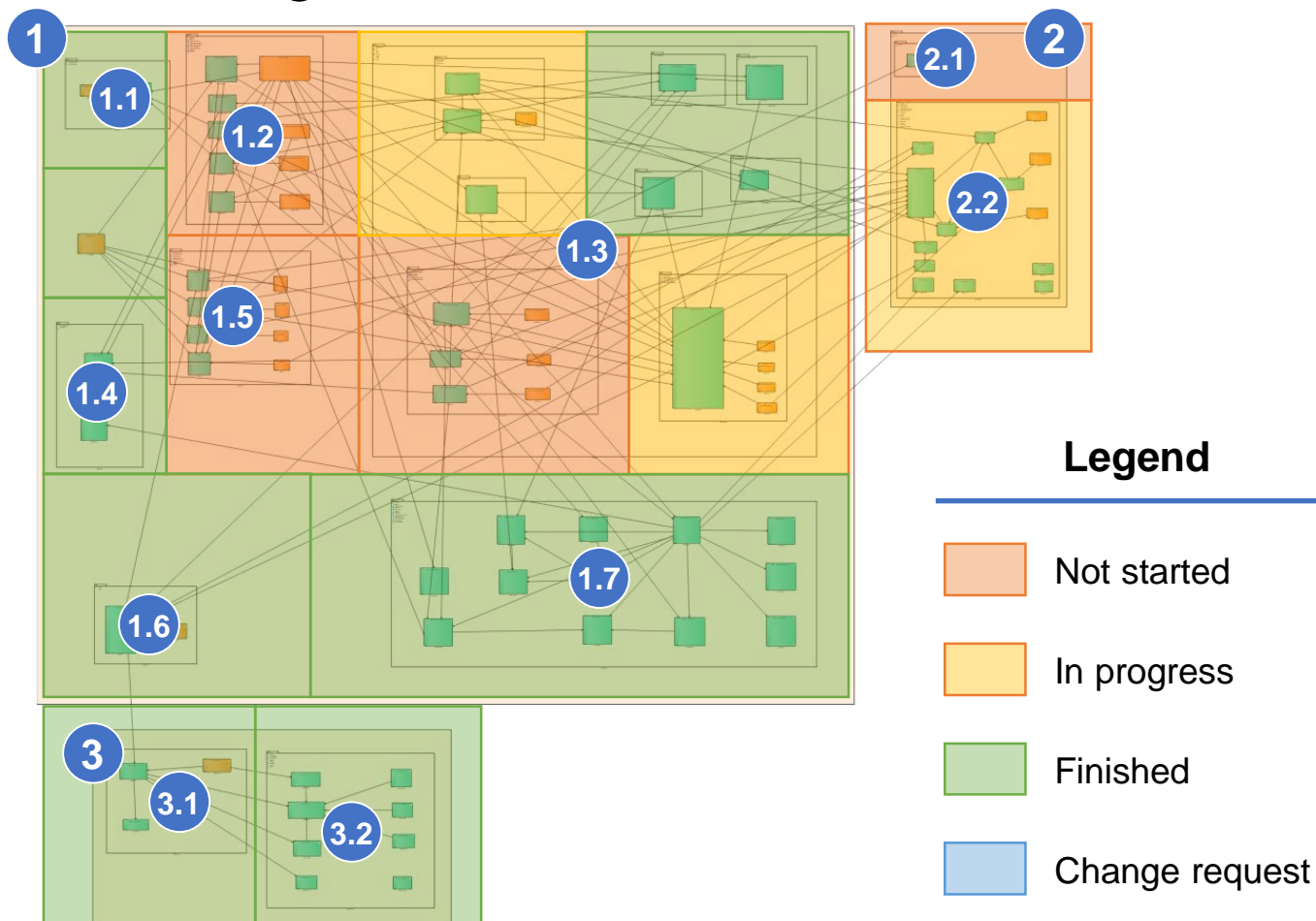
# Full Project View: Starting point



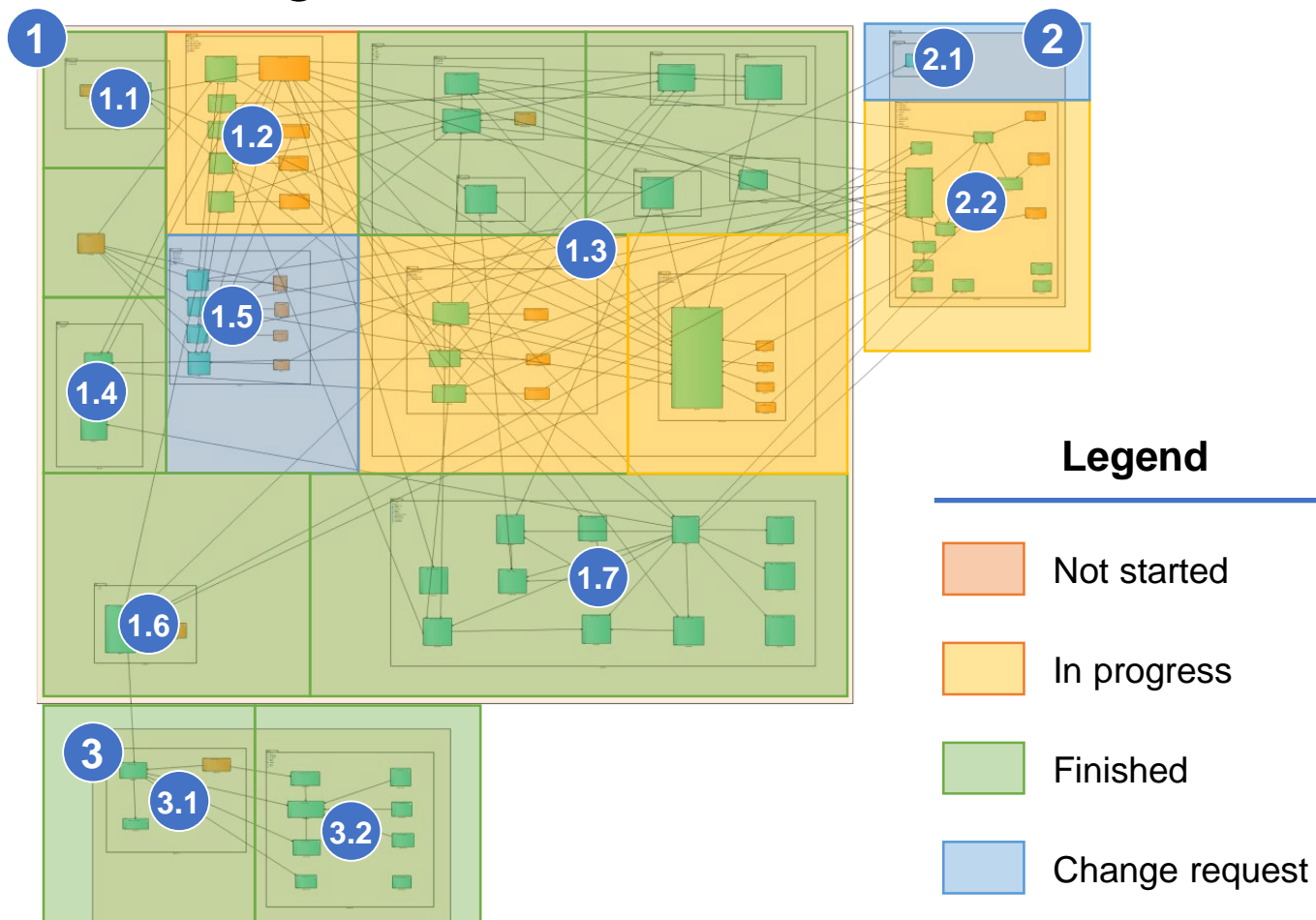
# Full Project View: 16.3.2017



# Full Project View: 20.4.2017

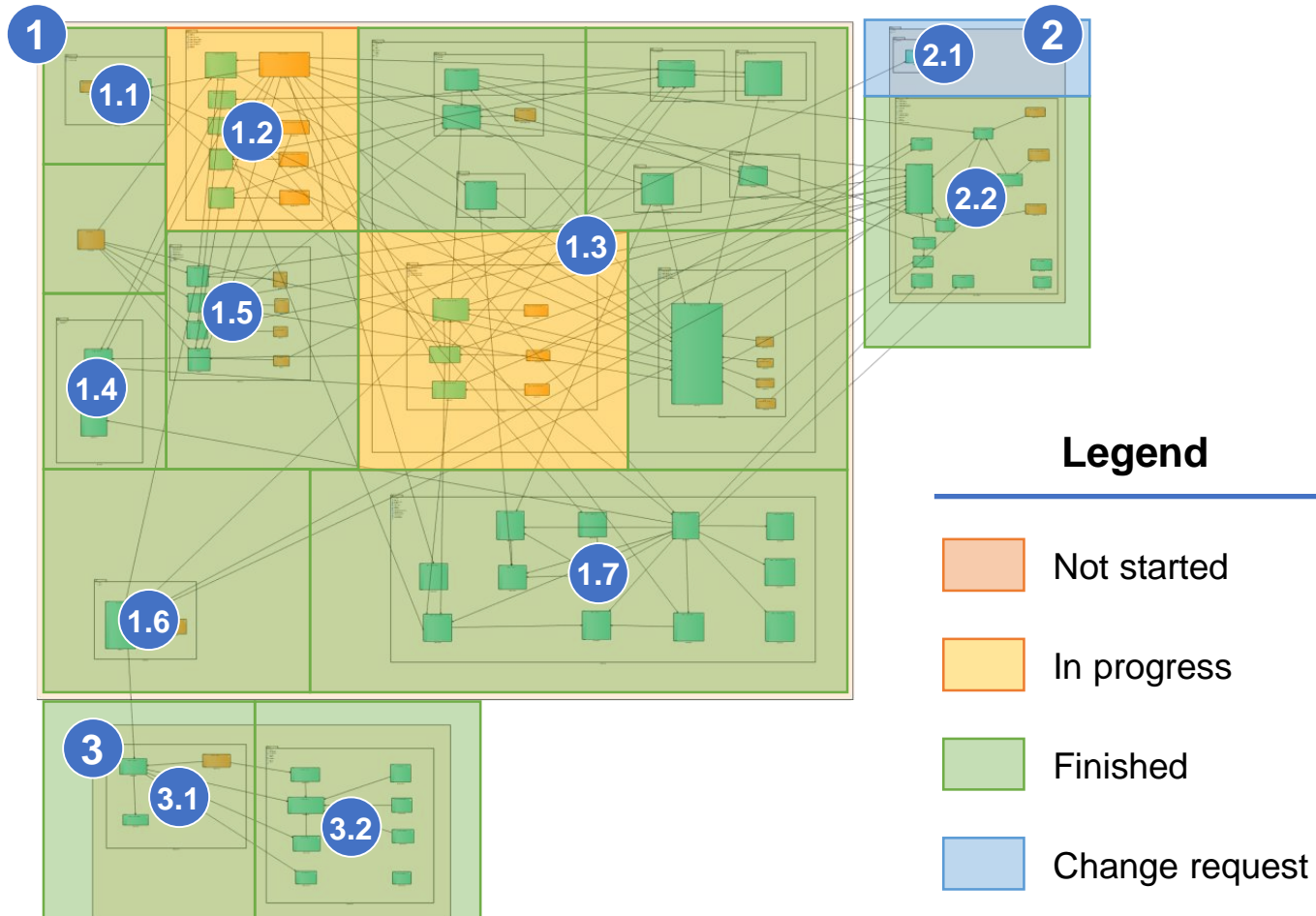


# Full Project View: 4.5.2017





# Full Project View: Current status





## Development plan:

Týden → Modul ↓	5	6	7	8	9	10	11	12	13	14	Responsible
Search											Miroslav Rudišin
Space						SAT needed					
State											
Utility											Jan Kohout
posix											
Cost											
Filter				Annealing							
Heuristic											
Report											
Goal											
IO						SAT needed					
Bookkeeping											
Events											David Löffler
sat/minisat -----											
Annealing -----											



## Development plan - reality:

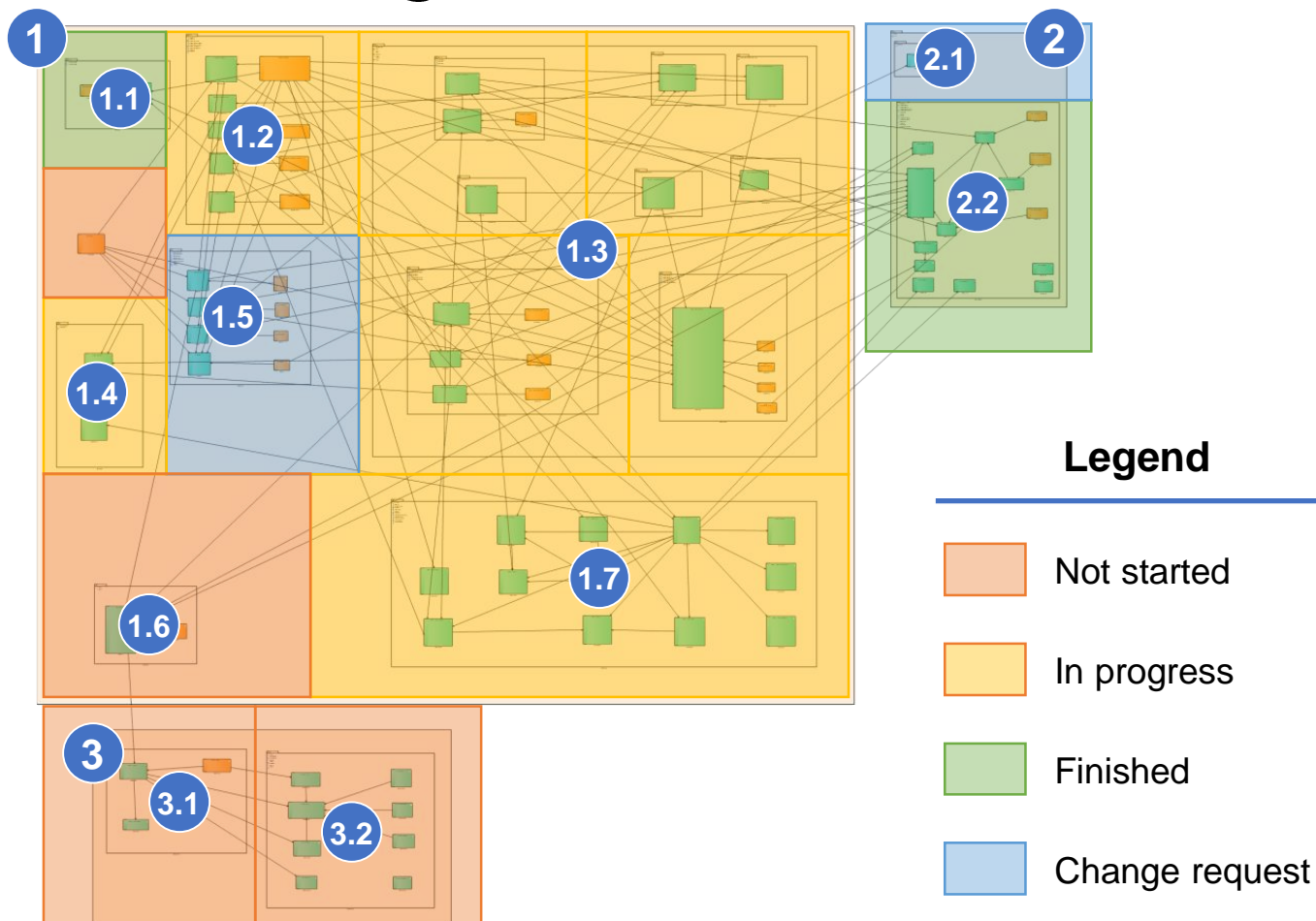
Week → Module ↓	5	6	7	8	9	10	11	12	13	14	Responsible
Search											Miroslav Rudišin
Space							SAT needed				David Löffler, Jan Kohout
State											Miroslav Rudišin, Kryštof Sýkora, David Löffler
Utility						overdue					Jan Kohout, David Löffler, Miroslav Rudišin
posix											David Löffler, Miroslav Rudišin
Cost											David Löffler, Miroslav Rudišin
Filter					Annealing						David Löffler
Heuristic											David Löffler
Report											Yevgeniya Chekh
Goal											David Löffler
IO									SAT needed		ChR by David Löffler
Bookkeeping											Yevgeniya Chekh, David Löffler
Events											David Löffler
sat/minisat											David Löffler
Annealing						Can be replaced					David Löffler
Binary search											Jan Kohout

# Deep Dive: Testing

# Testing

- **Input x output comparison**
- **Unit testing for development support**
- **Complications:**
  - **Could not build system**
  - **Unable to test beyond a certain degree of focus**
- **Further testing and strategies**

# Unit testing overview (4.5. 2017)



# Final individual reviews, Lessons learned

# Project evaluation: Marek Szeles

Team Role	Name	Contact
<b>Project Lead</b>	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# Project evaluation: Ho Minh Thanh

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# Project evaluation: David Löffler

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz



# Project evaluation: Kryštof Sýkora

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# Project evaluation: Miroslav Rudišín

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišín	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# Project evaluation: Jan Kohout

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# Project evaluation: Yevgeniya Chekh

Team Role	Name	Contact
Project Lead	Marek Szeles	szelemar@fel.cvut.cz
Analyst	Ho Minh Thanh	homintha@fel.cvut.cz
Head of Development	David Löffler	loffldav@fel.cvut.cz
Head of Testing	Kryštof Sýkora	sykorkry@fel.cvut.cz
Developer	Miroslav Rudišin	rudismi1@fel.cvut.cz
Developer	Jan Kohout	kohouj13@fel.cvut.cz
Tester	Yevgeniya Chekh	chekhyev@fel.cvut.cz

# Lessons learned

-  **Communication is key**
-  **Change requests by client are chaotic > gotta own the problem**
-  **Even the best analysis sometimes is not good enough**
-  **Even the best risk management sometimes does not help**
-  **It helps to have someone dedicated to information management**

## Points distribution

Team Role	Name	Points
Project Lead	Marek Szeles	25
Analyst	Ho Minh Thanh	25
Head of Development	David Löffler	25
Head of Testing	Kryštof Sýkora	25
Developer	Miroslav Rudišin	20
Developer	Jan Kohout	25
Tester	Yevgeniya Chekh	25

**Thank you for your attention.**

**Do you have  
questions?**