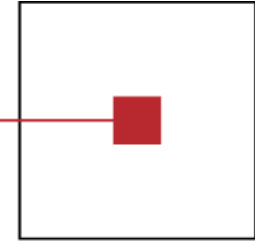




s c c h

software competence center  
hagenberg



# The Impact of Product Development on the Lifecycle of Defects

**Rudolf Ramler**

Software Competence Center Hagenberg, Austria

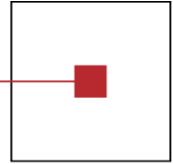
[rudolf.ramler@scch.at](mailto:rudolf.ramler@scch.at)

<http://www.scch.at>



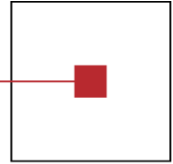
JOHANNES KEPLER  
UNIVERSITÄT LINZ

Netzwerk für Forschung, Lehre und Praxis

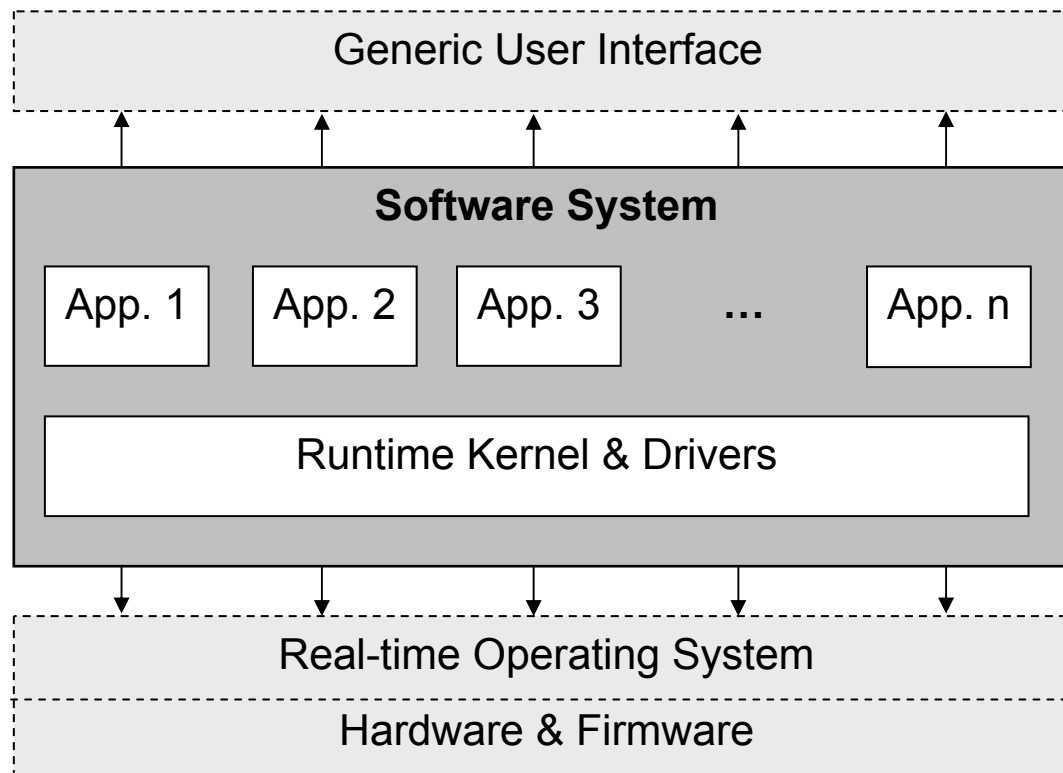


- Project Background
- Defect Lifecycle
- Product Development
- Defect Lifecycle - in Context of Product Development
- Impact of Product Development
- Summary & Further Questions



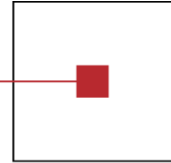


### ■ Embedded software system



# Project Background

Data from the studied project

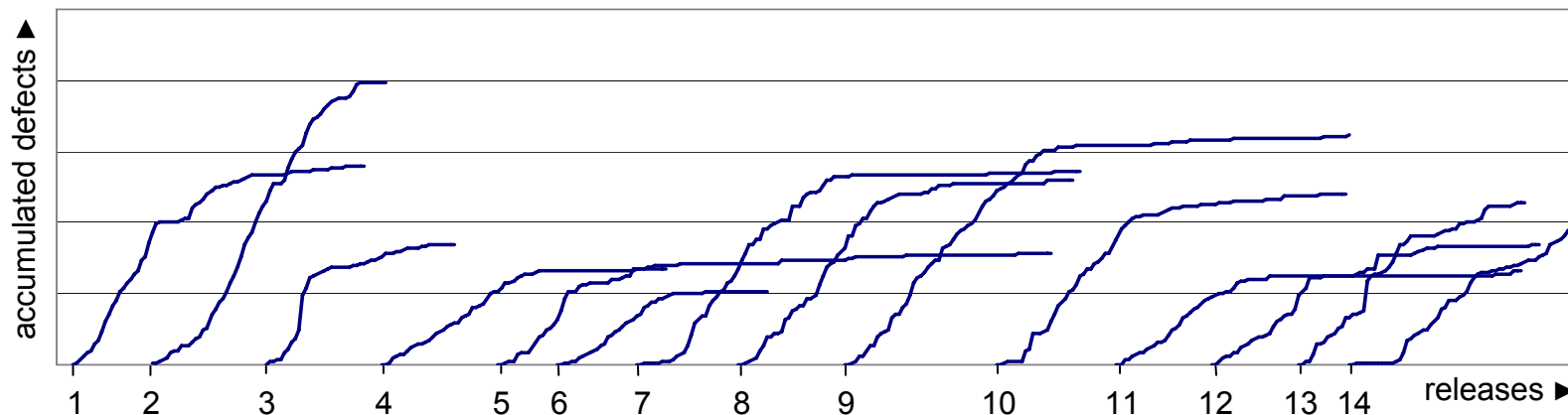


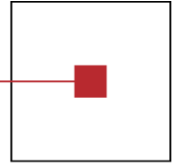
## ■ Issue repository

- 3.000+ resolved issue reports
- 68% defects, 32% enhancements
- This study considers only defects

## ■ Release database

- 130+ releases (main releases, integration releases, ...)
- This study: 14 consecutive releases on the main line (ca. 2 years)



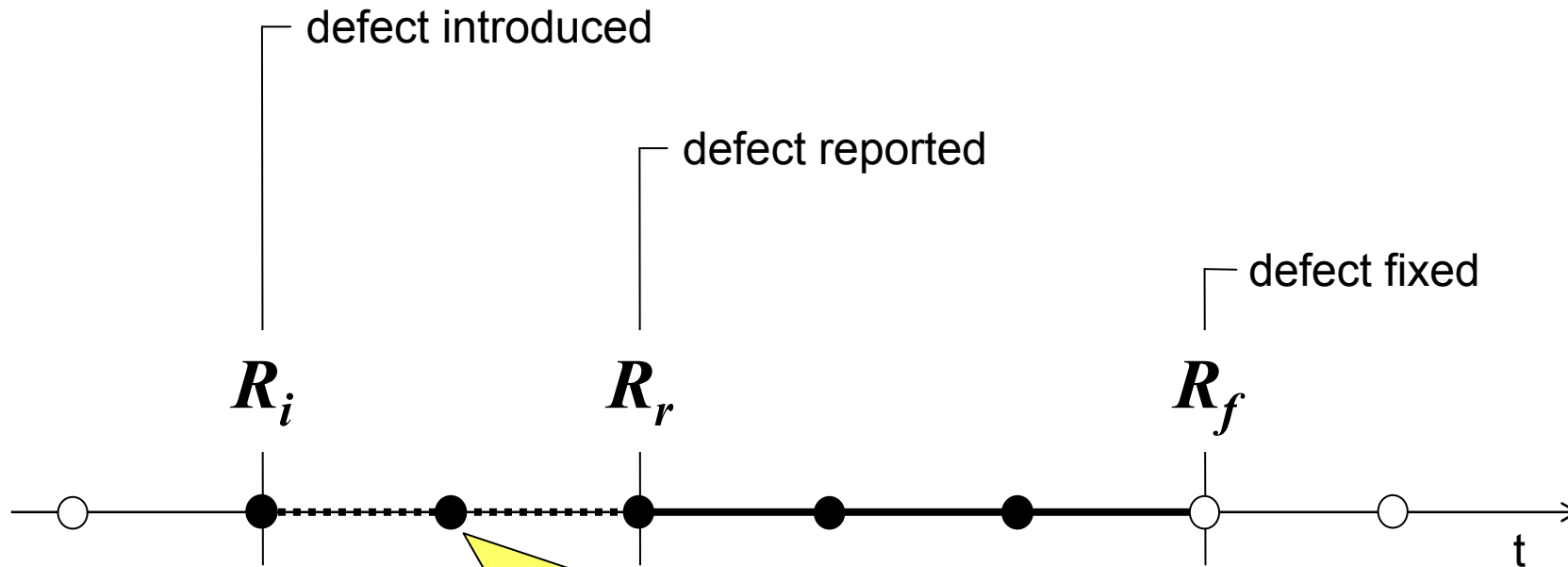
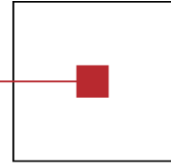


- Project Background
- Defect Lifecycle
- Product Development
- Defect Lifecycle - in Context of Product Development
- Impact of Product Development
- Summary & Further Questions



# Defect Lifecycle

## Overview of main points



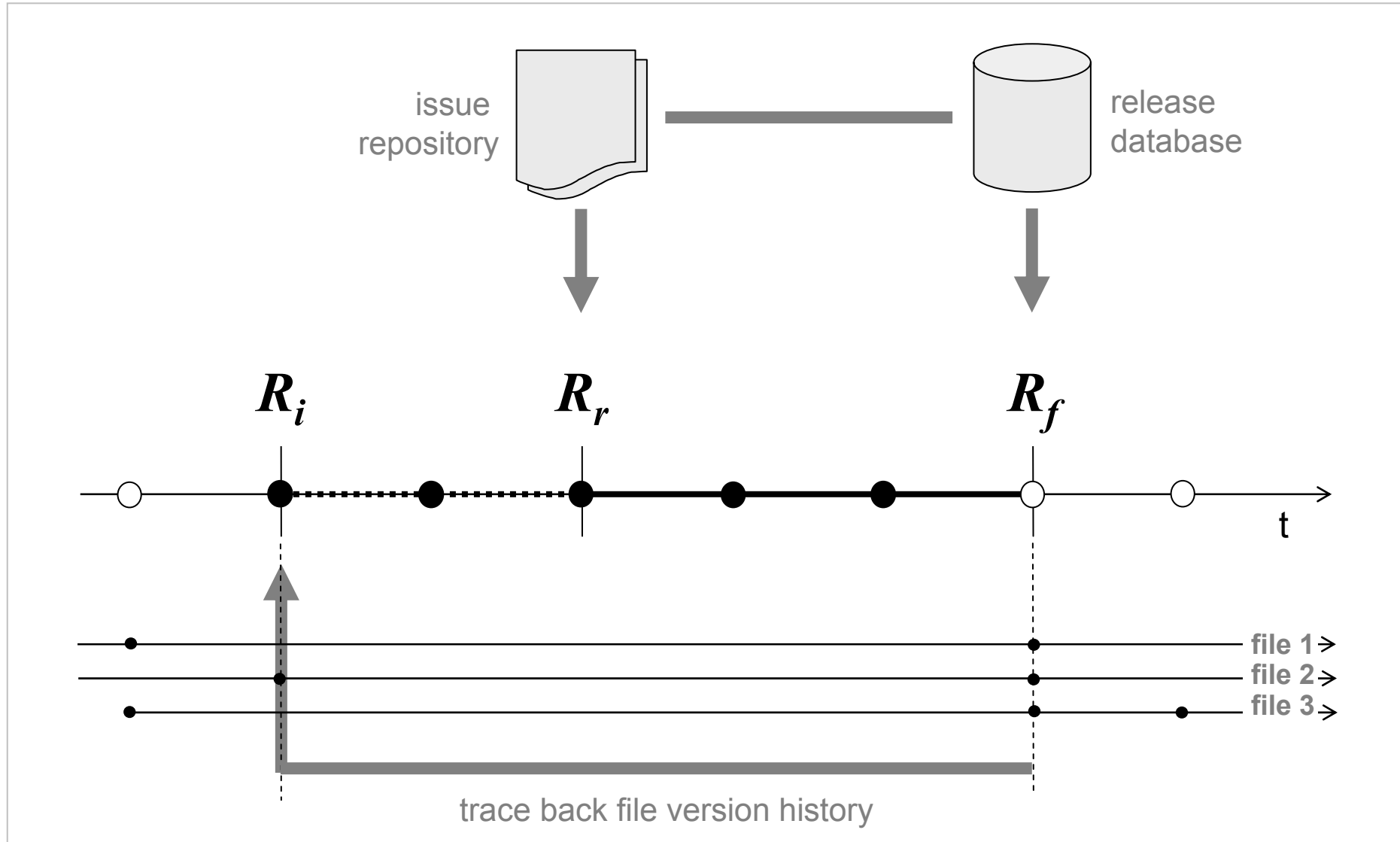
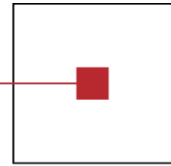
Why is it important to know open defects per release?  
→ Accurate historical data, e.g., to construct  
defect prediction models

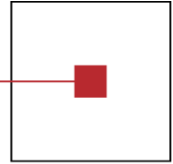
# Defect Lifecycle

Where does the data come from?

s c c h

software competence center  
hagenberg



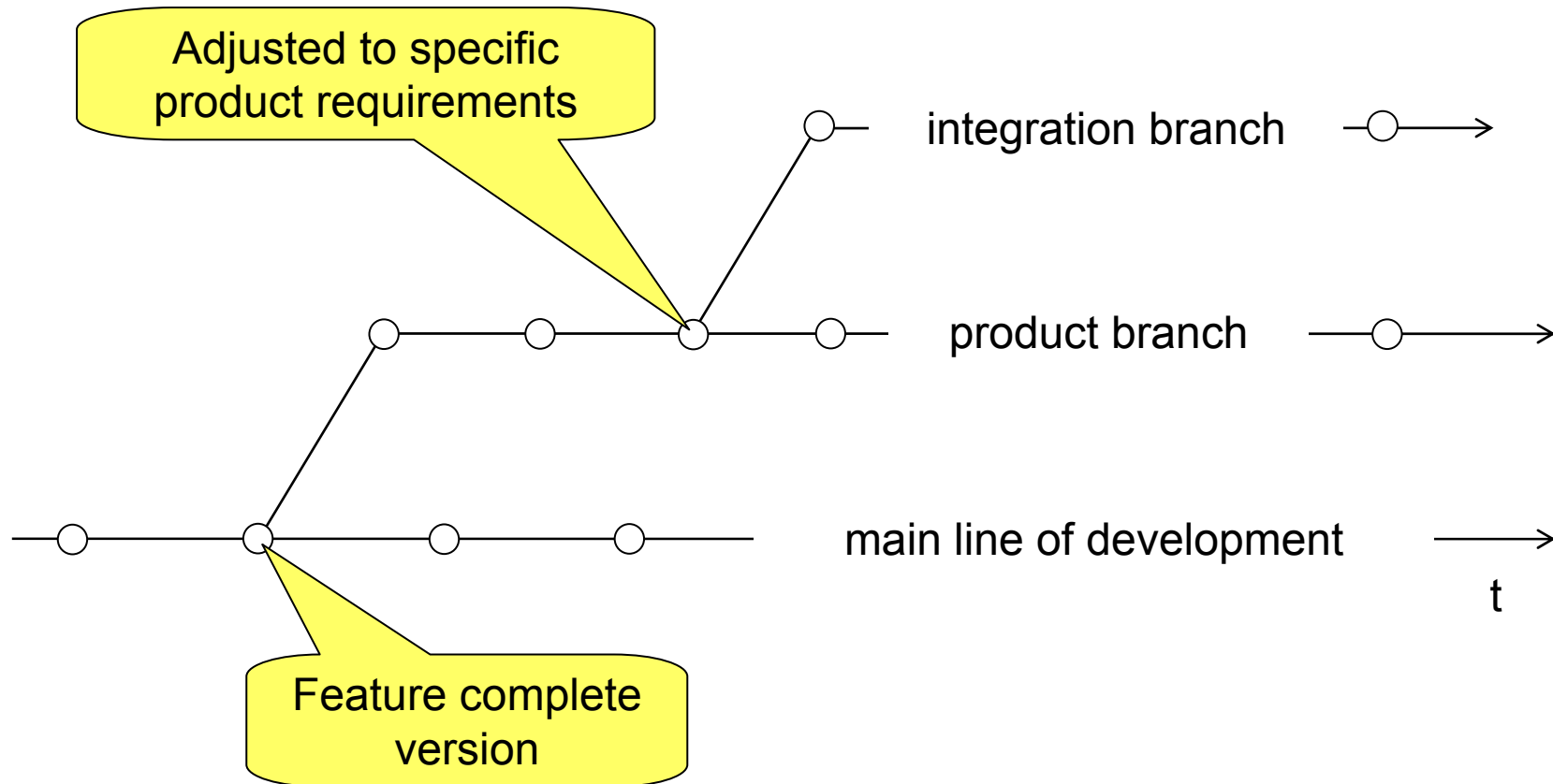
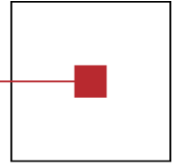


- Project Background
- Defect Lifecycle
- Product Development
- Defect Lifecycle - in Context of Product Development
- Impact of Product Development
- Summary & Further Questions



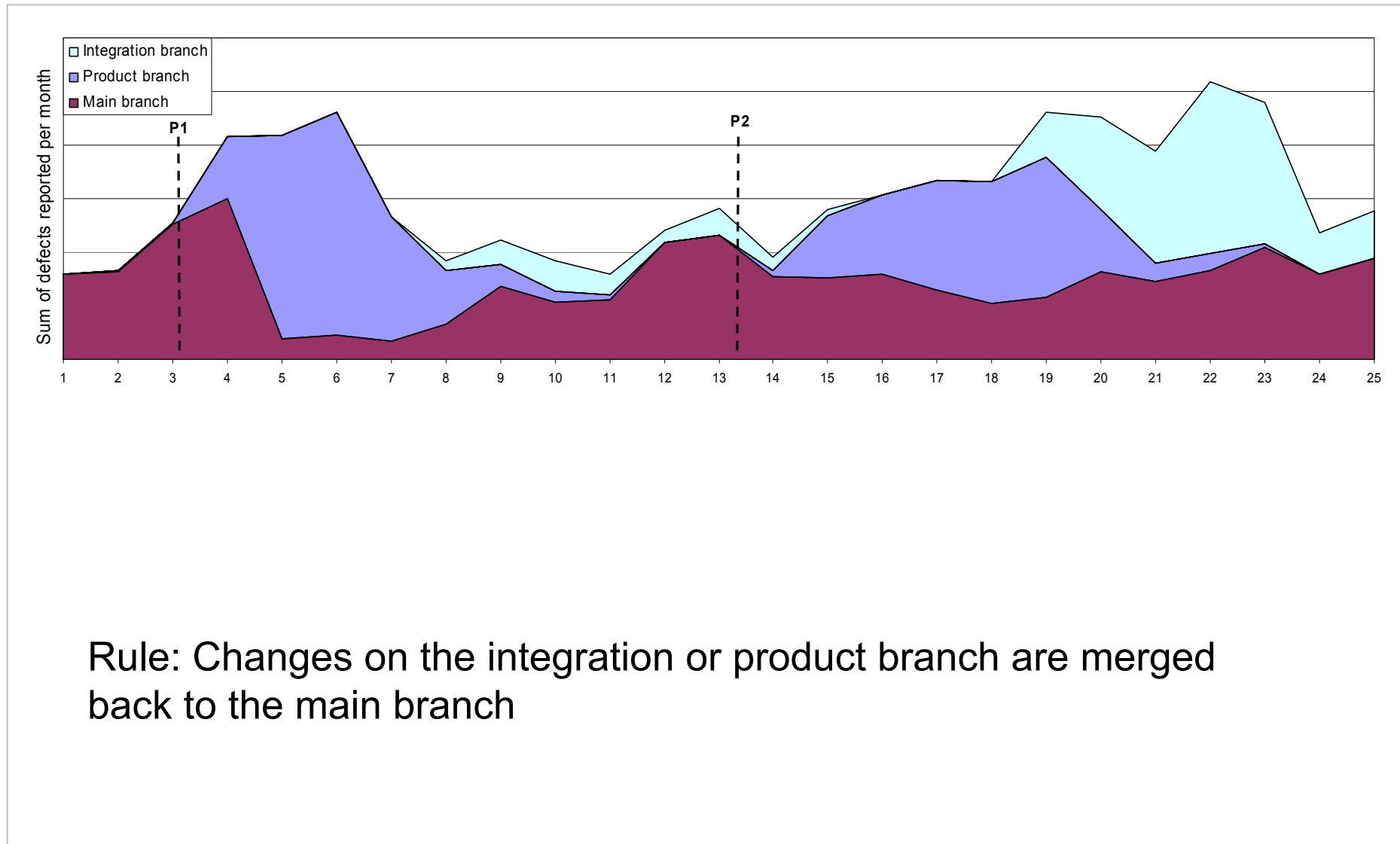
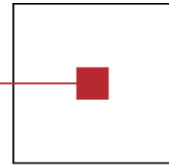
# Product Development

## Organization of software development

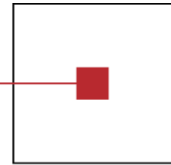


# Product Development

## Number defects on branches



Rule: Changes on the integration or product branch are merged back to the main branch



- Project Background
- Defect Lifecycle
- Product Development
- Defect Lifecycle - in Context of Product Development
- Impact of Product Development
- Summary & Further Questions

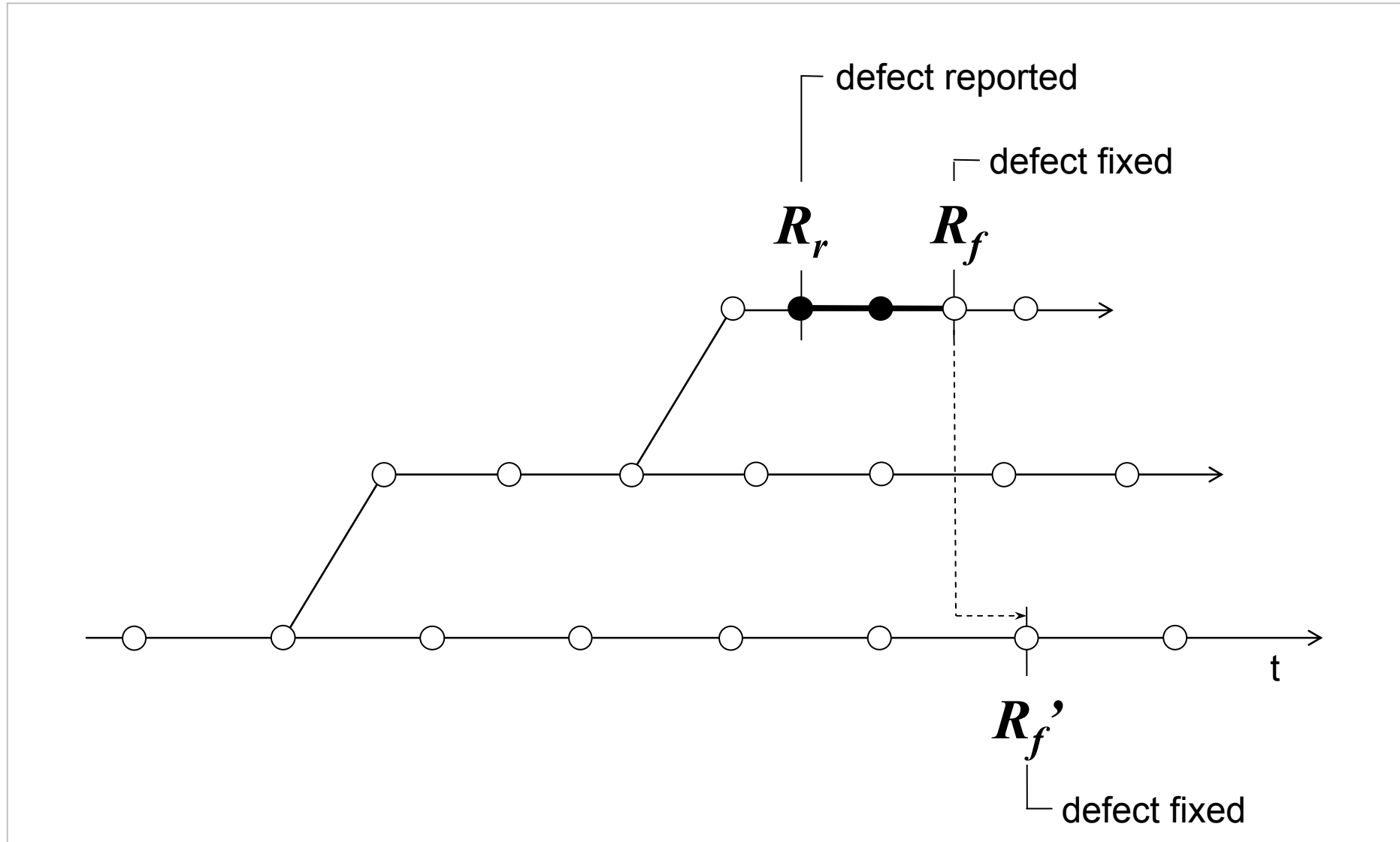
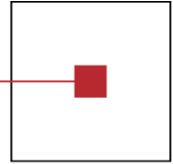


# Defect Lifecycle

... in context of product development

s c c h

software competence center  
hagenberg

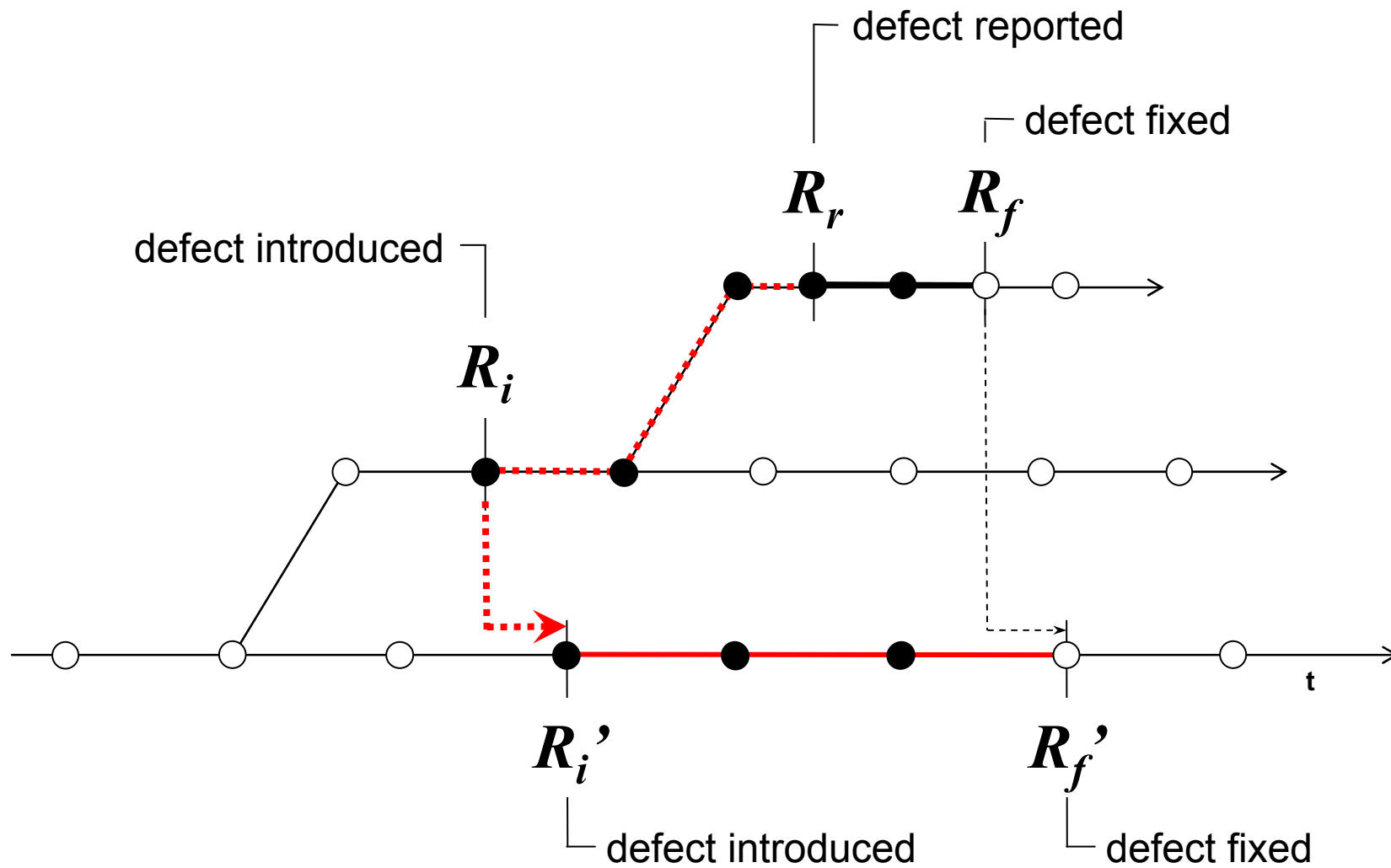
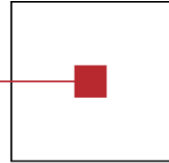


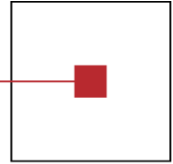
# Defect Lifecycle

... in context of product development

s c c h

software competence center  
hagenberg



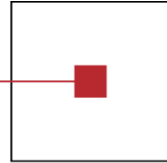


- Project Background
- Defect Lifecycle
- Product Development
- Defect Lifecycle - in Context of Product Development
- Impact of Product Development
- Summary & Further Questions



# Impact of Product Development

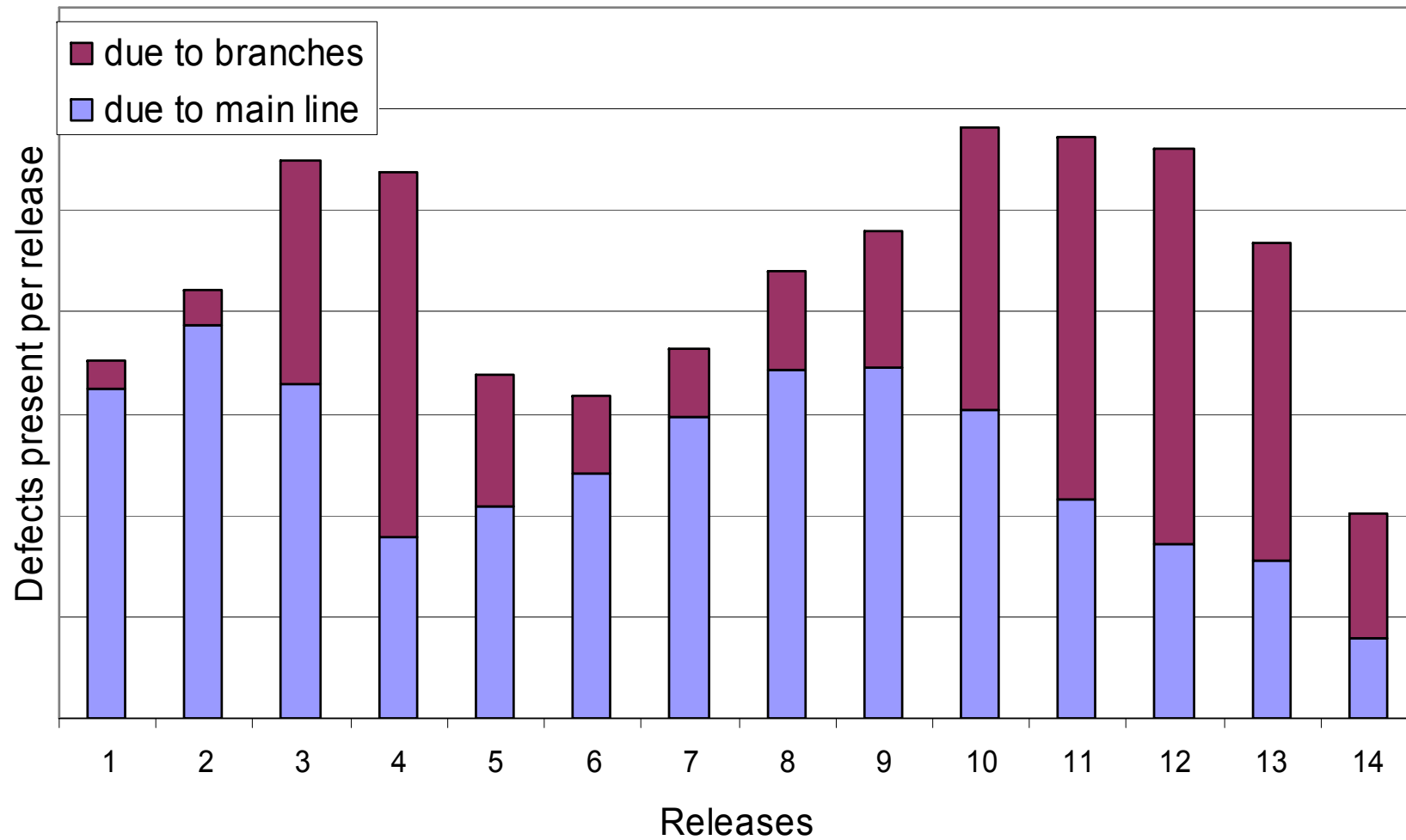
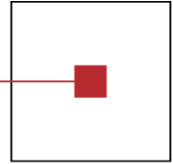
## Number of defects ...

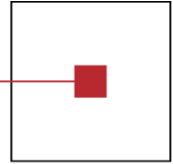


Release	Defects introduced	Defects reported	Defects fixed	Defects open
1	3%	0%	0%	8%
2	5%		0%	8%
3	58%		0%	40%
4	82%		57%	67%
5	44%		80%	38%
6	23%		64%	24%
7	15%		26%	19%
8	19%		3%	22%
9	23%		5%	28%
10	56%		12%	48%
11	72%		32%	62%
12	80%		57%	69%
13	63%		70%	67%
14	65%		70%	62%

# Impact of Product Development

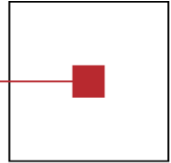
## Number of open defects per release





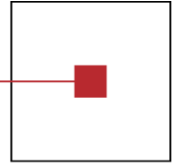
- Project Background
- Defect Lifecycle
- Product Development
- Defect Lifecycle - in Context of Product Development
- Impact of Product Development
- Summary & Further Questions





- Accurate historical data on open defects considered important, e.g., for creating defect prediction models
- Open defects were derived from the defect lifecycle
- Defects reported on product and integration branches impact the number of open defects on the main line
- Accurate historical data requires to include the defects reported on product and integration branches
- Product and integration branches are a consequence of the product development strategy

## Further Questions



- Is the presented case representative?
- How to cope with incomplete defect data (before product development and integration)?
- Can prediction work in this context at all?

**Thank you!**