TdSE'24

Enhancing MBSE Education with Version Control and Automated Feedback

Levente Bajczi[1], Dániel Szekeres [1], Daniel Siegl [2], Vince Molnár [1]

[1] Budapest University of Technology and Economics
Department of Artifical Intelligence and Systems Engineering

[2] LieberLieber Software GmbH

MŰEGYETEM 1782

ftsrg

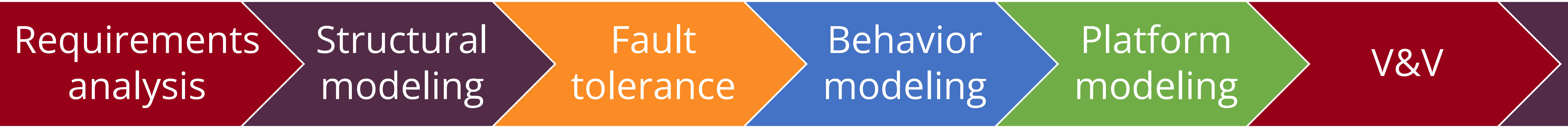
Critical Systems
Research Group

LieberLieber

MBSE Course at TU Budapest

- **Focus:** (Model-Based) Systems Engineering
  - Modeling primarily in SysML
  - Platform-driven methodology
- **Evaluations:**
  - Exam with multiple-choice and constructive tasks
  - **Homework:**
    - Groups of 3 → Collaborative modeling!
    - Complex SysML model in 6 steps

80-120 participants



Requirements analysis

Structural modeling

Fault tolerance

Behavior modeling

Platform modeling

V&V

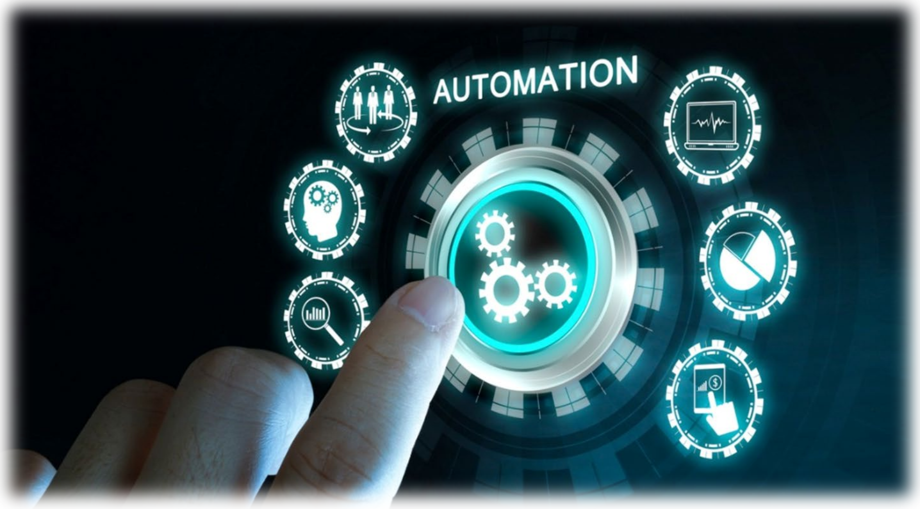
Motivation & History

- **Challenges:**
  - Collaborative modeling
  - Managing tasks and submissions
  - Grading 30-40 projects 6 times per semester
- Long-time vision: **Automated workflows**
  - For collaboration and feedback
- Previous editions:
  - Modeling in MagicDraw; Version control on Teamwork Server; Documentation and grading in Moodle
  - Modeling in Papyrus; Version control and documentation on GitHub; Grading in Moodle

Requirements

- **Goals:**
  1. Early & automated feedback
  2. Issue and progress tracking
  3. Integrated and always up-to-date documentation

  Summarized: DevOps for SysML

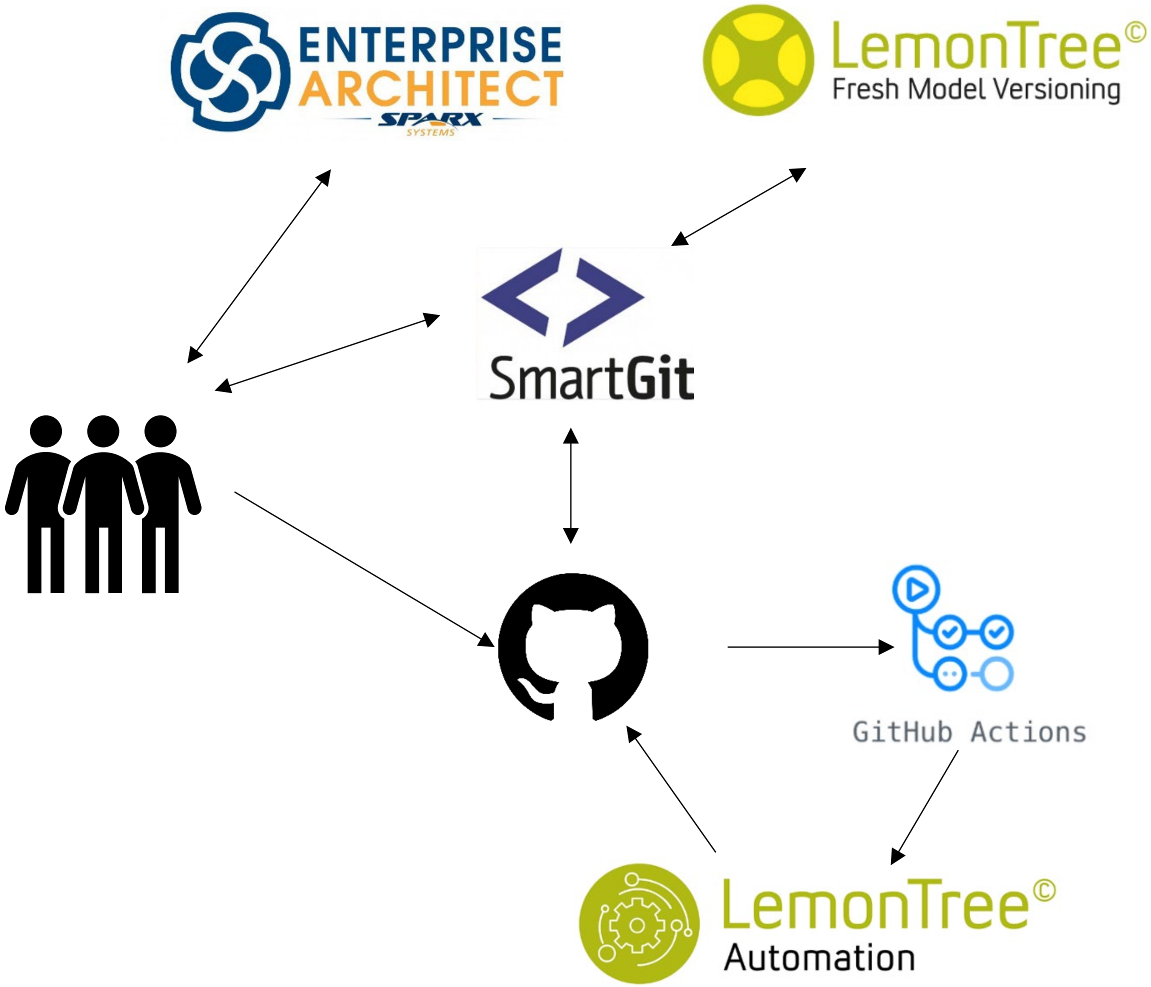


- Agile feedback loops
- Make learning/teaching systems engineering less frustrating
- Motivate students to learn (and get familiar with modern tools)

Toolchain

Inspired by **best practices** in **software engineering**

- Development on **feature branches**
- Convenient **graphical diff/merge** from Git client and from pull requests
- **Automation** in GitHub Actions
  - Using private runners
    - GDPR, Licence & IP protection



Homework Assignment Details

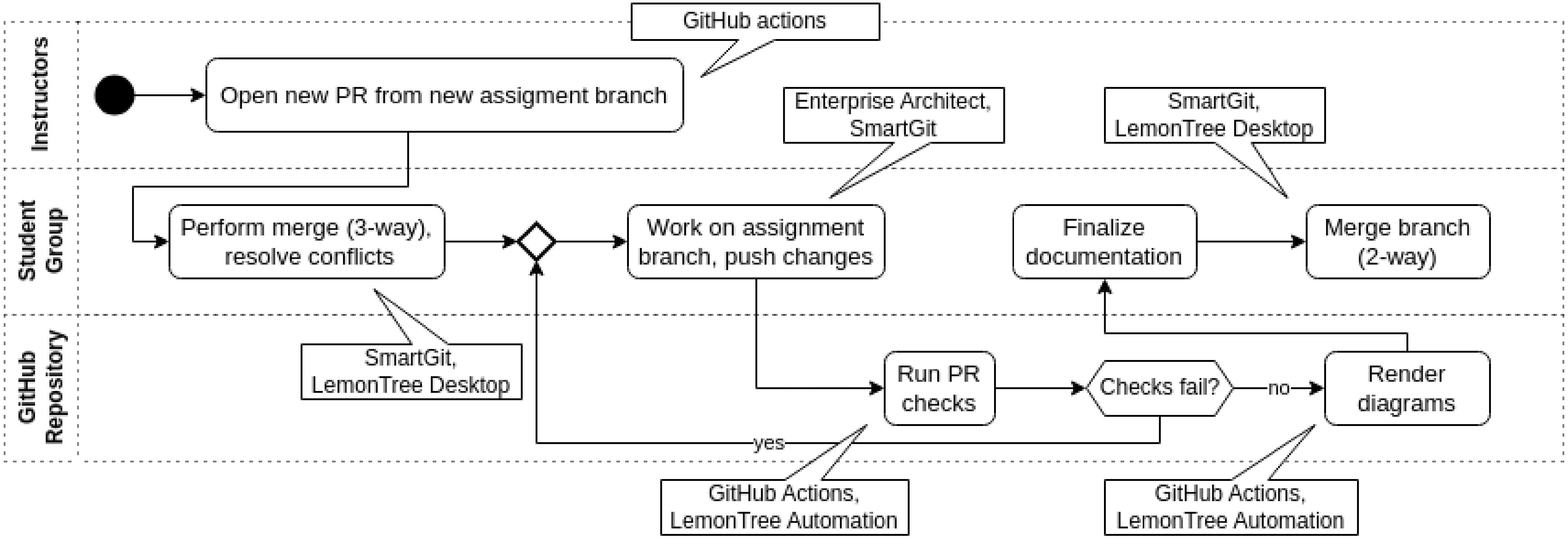
Design of an “IntelliBus” System

- Autonomous buses providing on-demand transportation in a closed office park

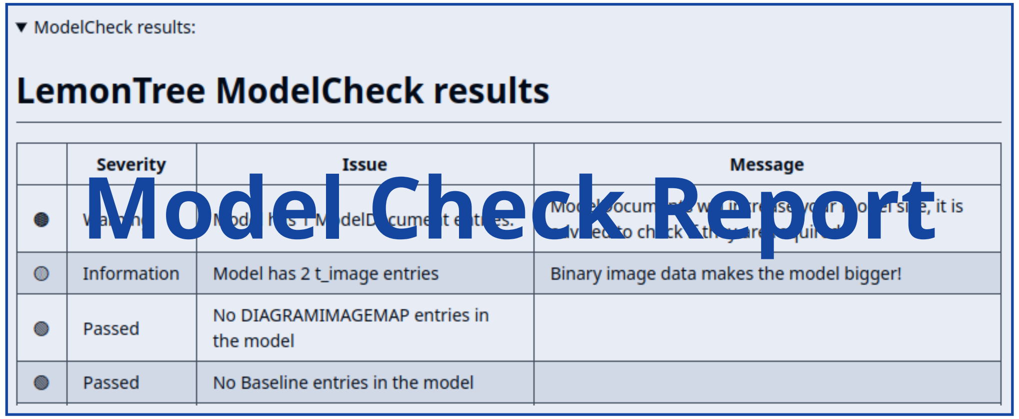
1. **Requirement** engineering for the whole system
2. **Structural** modeling: traceability to requirements, top-down functional decomposition, bottom-up platform design
3. **Safety** analysis of an adaptive cruise control (ACC) component
4. **Behavior** modeling: detailed design of the boarding equipment
5. **Platform** allocation and architecture for the ACC component
6. **V&V** for the ACC component (requirement review, test definition)

Workflow

Assignment: Opening a PR
Submission: Merging a PR
Documentation: Markdown with dynamically updating diagram renders
Evaluation: Automated checks for WiP, ***then*** human feedback


Instructors
Student Group
GitHub Repository
GitHub actions
Open new PR from new assigment branch
Enterprise Architect, SmartGit
SmartGit, LemonTree Desktop
Perform merge (3-way), resolve conflicts
Work on assignment branch, push changes
Finalize documentation
Merge branch (2-way)
SmartGit, LemonTree Desktop
Run PR checks
Checks fail?
no
Render diagrams
yes
GitHub Actions, LemonTree Automation
GitHub Actions, LemonTree Automation

Example Feedback

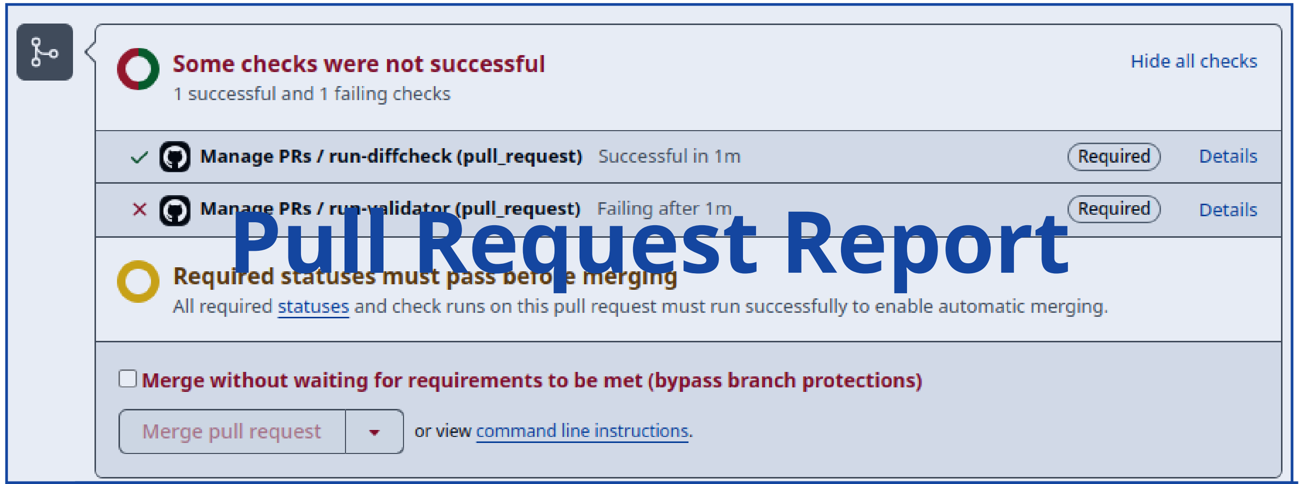


▼ ModelCheck results:

LemonTree ModelCheck results

	Severity	Issue	Message
●	Warning	Model has 1 ModelDocument entries.	ModelDocuments will increase your model size, it is advised to check if they are required.
○	Information	Model has 2 t_image entries	Binary image data makes the model bigger!
●	Passed	No DIAGRAMIMAGEMAP entries in the model	
●	Passed	No Baseline entries in the model	

Model Check Report



Some checks were not successful

1 successful and 1 failing checks

Hide all checks

✓ **Manage PRs / run-diffcheck (pull_request)** Successful in 1m Required Details

✕ **Manage PRs / run-validator (pull_request)** Failing after 1m Required Details

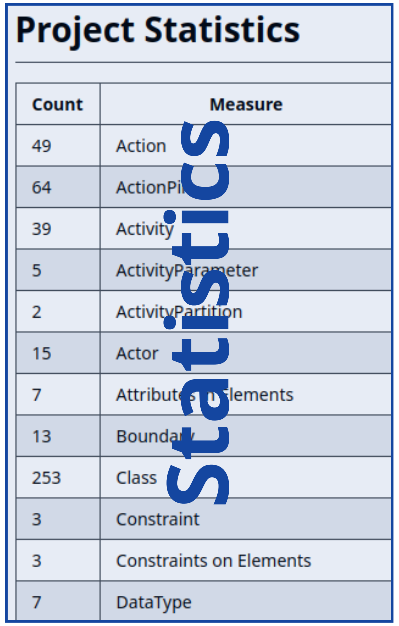
Required statuses must pass before merging

All required statuses and check runs on this pull request must run successfully to enable automatic merging.

□ **Merge without waiting for requirements to be met (bypass branch protections)**

or view command line instructions.

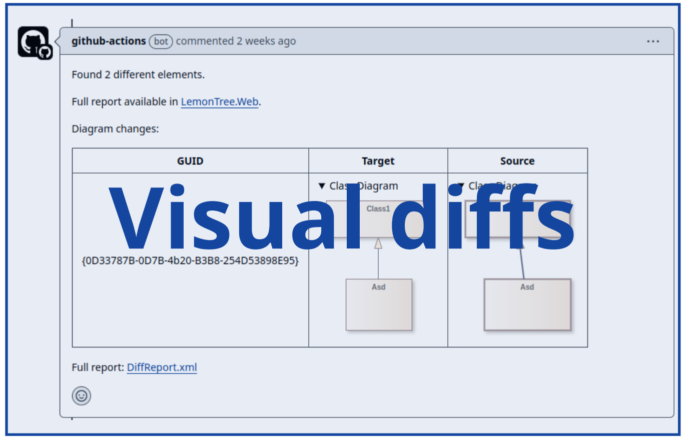
Pull Request Report



Project Statistics

Count	Measure
49	Action
64	ActionPin
39	Activity
5	ActivityParameter
2	ActivityPartition
15	Actor
7	Attributes on Elements
13	Boundary
253	Class
3	Constraint
3	Constraints on Elements
7	DataType

Statistics



github-actions (bot) commented 2 weeks ago

Found 2 different elements.

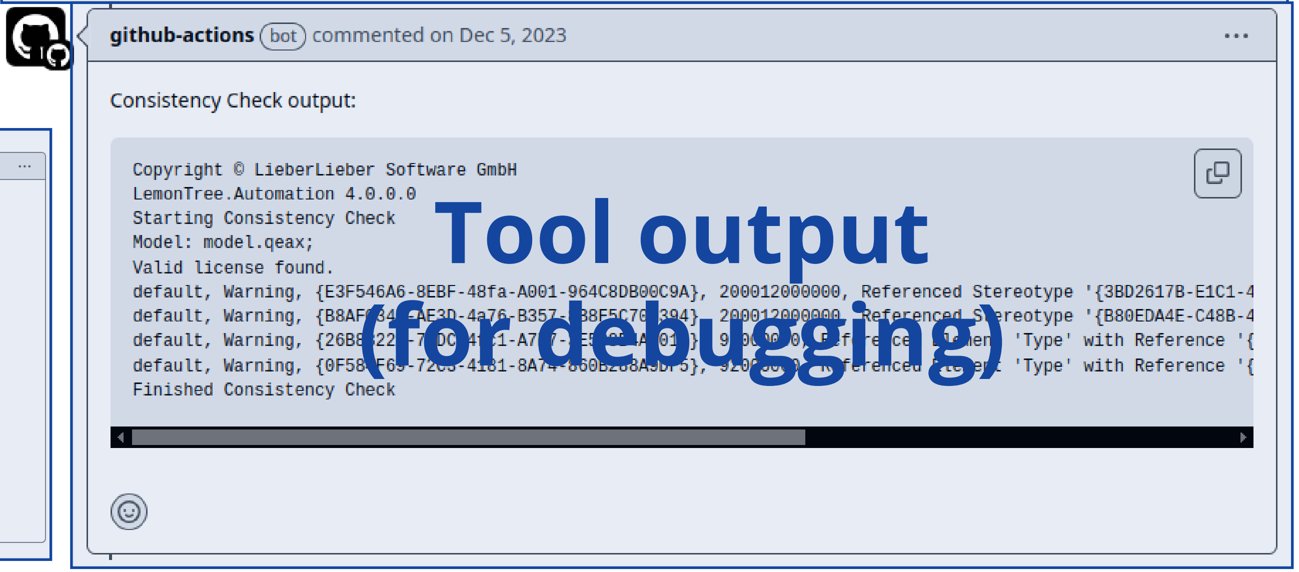
Full report available in LemonTree.Web.

Diagram changes:

GUID	Target	Source
{0D33787B-0D7B-4b20-B3B8-254D53898E95}	▼ Class Diagram	▼ Class Diagram

Full report: DiffReport.xml

Visual diffs



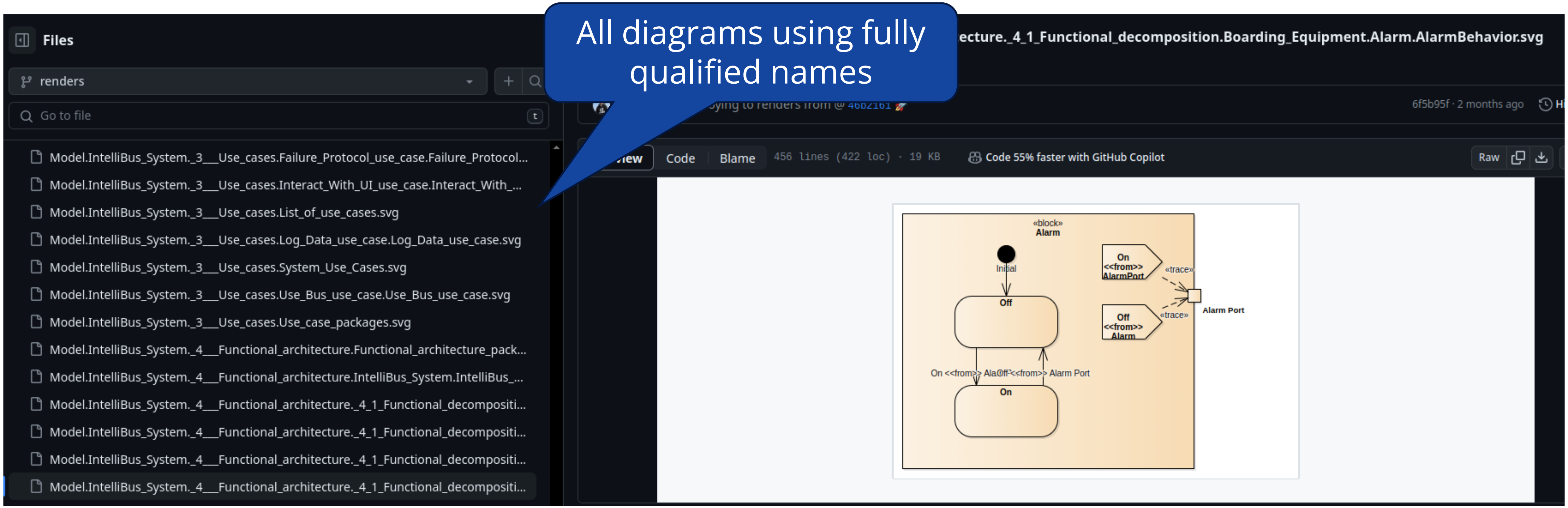
github-actions (bot) commented on Dec 5, 2023

Consistency Check output:

```
Copyright © LieberLieber Software GmbH
LemonTree.Automation 4.0.0.0
Starting Consistency Check
Model: model.qeax;
Valid license found.
default, Warning, {E3F546A6-8EBF-48fa-A001-964C8DB00C9A}, 200012000000, Referenced Stereotype '{3BD2617B-E1C1-4
default, Warning, {B8AF[illegible]34-AE3D-4a76-B357-[illegible]38F5C70[illegible]394}, 200012000000, Referenced Stereotype '{B80EDA4E-C48B-4
default, Warning, {26B8[illegible]22[illegible]-7[illegible]DC[illegible]4[illegible]1-A7[illegible]7-[illegible]E5[illegible]A[illegible]01[illegible]}, 9[illegible]0, Reference[illegible] Elemen[illegible] 'Type' with Reference '{
default, Warning, {0F58[illegible]F6[illegible]-72[illegible]-4181-8A74-860B[illegible]}, 92[illegible], Referenced Element 'Type' with Reference '{
Finished Consistency Check
```

Tool output (for debugging)

Auto-Updating Documentation



All diagrams using fully qualified names

Files

renders

Go to file

- Model.IntelliBus_System._3__Use_cases.Failure_Protocol_use_case.Failure_Protocol...
- Model.IntelliBus_System._3__Use_cases.Interact_With_UI_use_case.Interact_With_...
- Model.IntelliBus_System._3__Use_cases.List_of_use_cases.svg
- Model.IntelliBus_System._3__Use_cases.Log_Data_use_case.Log_Data_use_case.svg
- Model.IntelliBus_System._3__Use_cases.System_Use_Cases.svg
- Model.IntelliBus_System._3__Use_cases.Use_Bus_use_case.Use_Bus_use_case.svg
- Model.IntelliBus_System._3__Use_cases.Use_case_packages.svg
- Model.IntelliBus_System._4__Functional_architecture.Functional_architecture_pack...
- Model.IntelliBus_System._4__Functional_architecture.IntelliBus_System.IntelliBus_...
- Model.IntelliBus_System._4__Functional_architecture._4_1_Functional_decompositi...
- Model.IntelliBus_System._4__Functional_architecture._4_1_Functional_decompositi...
- Model.IntelliBus_System._4__Functional_architecture._4_1_Functional_decompositi...
- Model.IntelliBus_System._4__Functional_architecture._4_1_Functional_decompositi...

ecture._4_1_Functional_decomposition.Boarding_Equipment.Alarm.AlarmBehavior.svg

ying to renders from @ 46b2161

6f5b95f · 2 months ago

iew | Code | Blame | 456 lines (422 loc) · 19 KB

Code 55% faster with GitHub Copilot

Raw

Can easily be used in markdown documentation!

```
![diagram](../../raw/renders/Model.IntelliBus_System._3___Use_cases.System_use_cases.svg)
```

DEMO

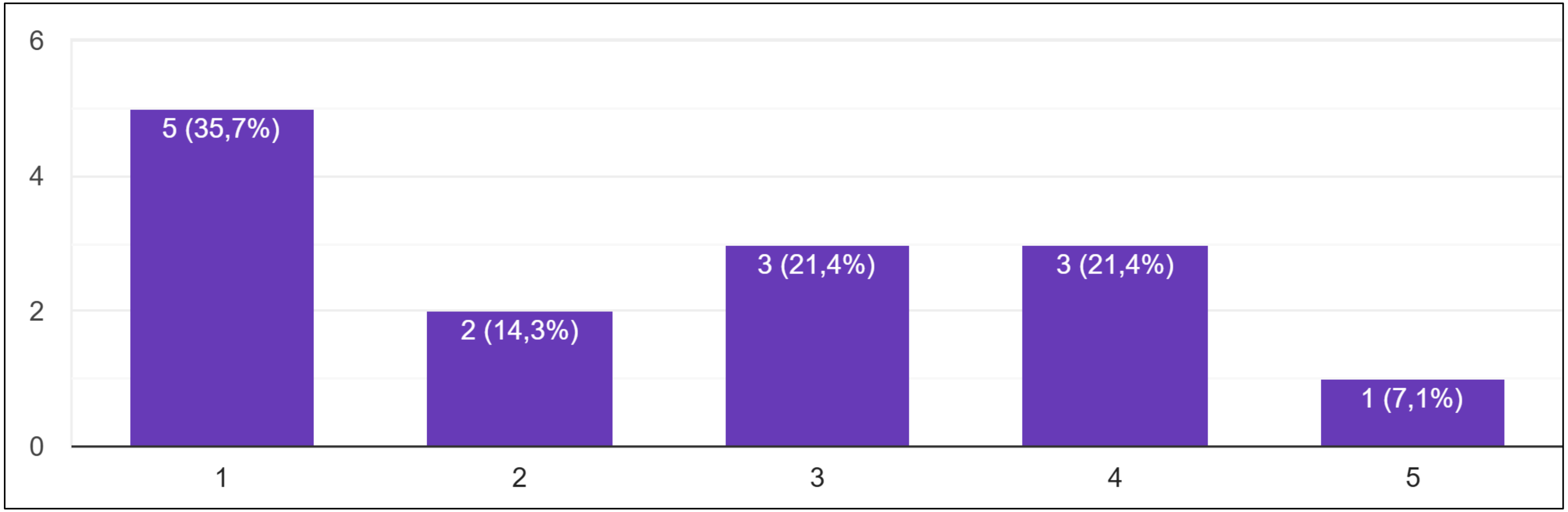
Evaluation with Students

- Course of 2023/24/1 (Autumn)
  - **Tutorial:**
    - Setting up the tools and performing simple 2-way and 3-way merging
  - **Homework:**
    - Initial repository with Assignment 1
    - 5 subsequent assignments via automated PRs
    - Any 2 assignments may be skipped, but incentives for not skipping

Student Feedback

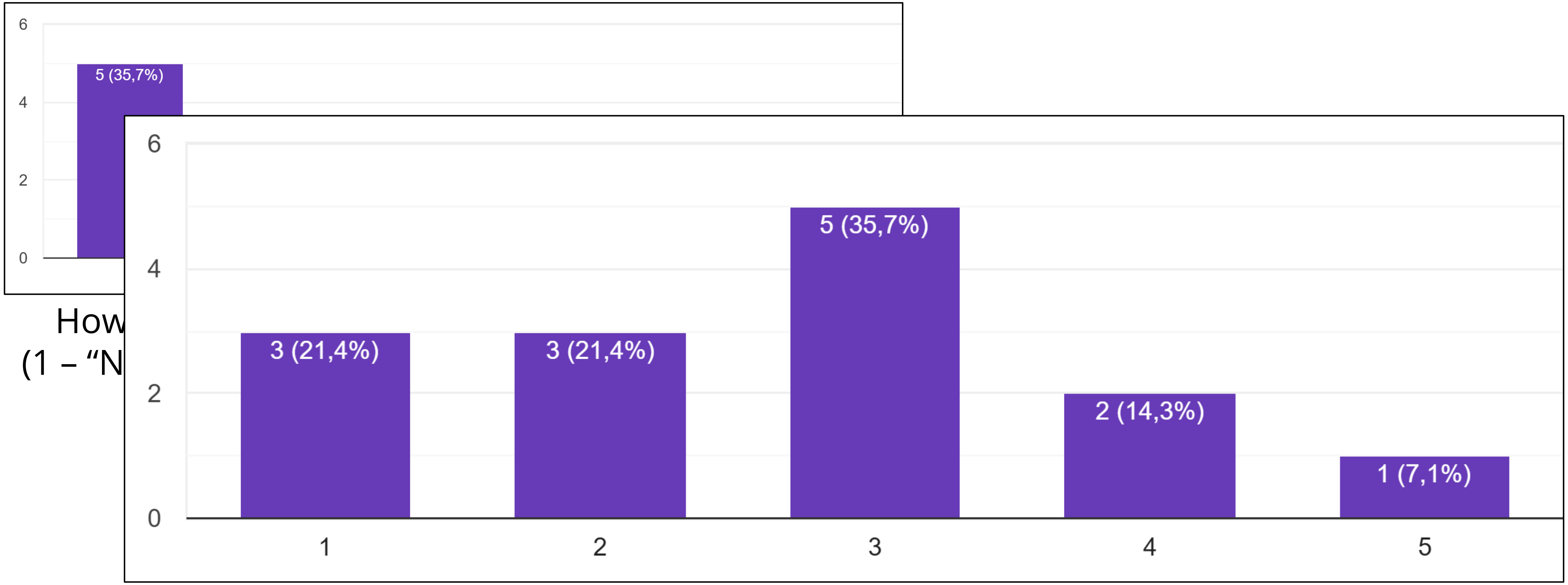
- Students **appreciated LemonTree** as a standalone tool
  - But had a hard time working with merge conflicts in general
  - Previous years: typical solution was to discard one of versions
- "Model merging **should be emphasized** in the course material and the optional tutorial should be mandatory"
  - Using tools efficiently requires training
  - Even if the tool is efficient and intuitive
- **79% completion rate** of the homework
  - Typical level in the past years

Student Feedback



How satisfied were you with Enterprise Architect?
(1 – “Never want to use it again”; 5 – “It was excellent”)

Student Feedback

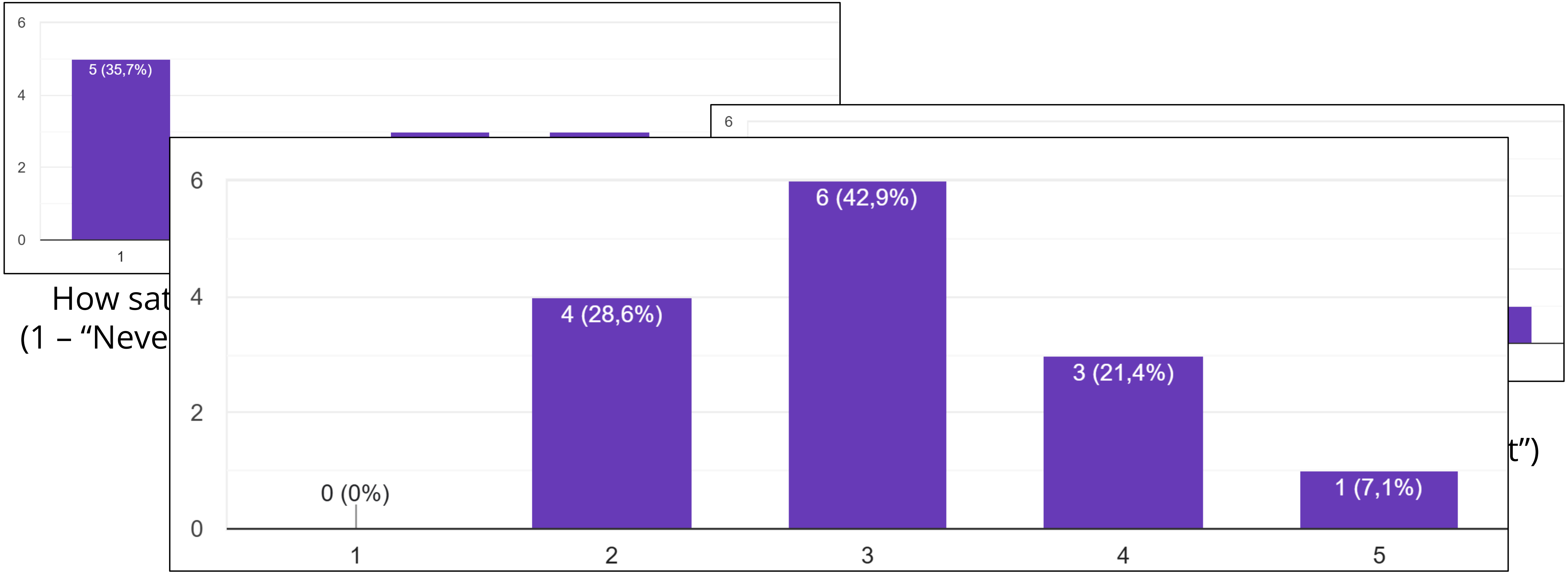


How

(1 – “N

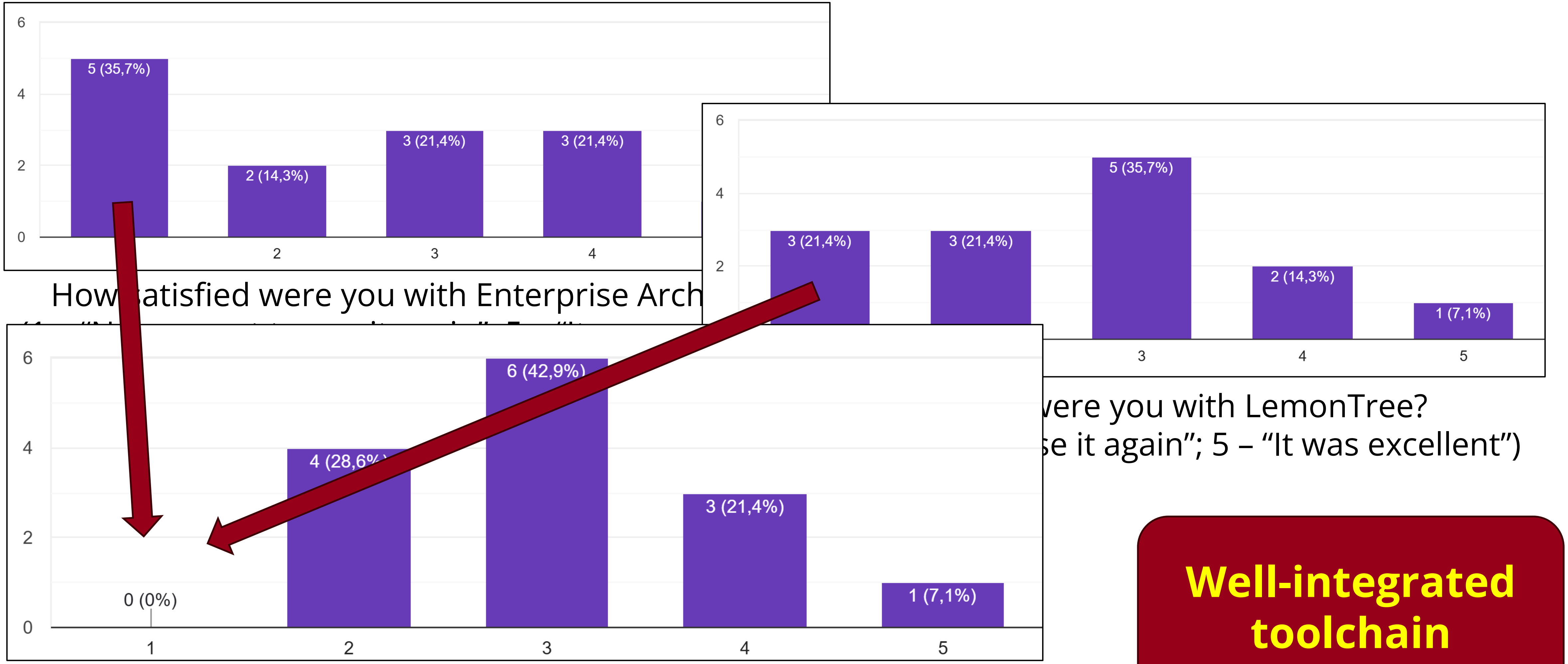
How satisfied were you with LemonTree?
(1 – “Never want to use it again”; 5 – “It was excellent”)

Student Feedback



How satisfied were you with the whole infrastructure (EU, LT, GH)?
(1 – “Never want to use it again”; 5 – “It was excellent”)

Student Feedback



How satisfied were you with Enterprise Arch

were you with LemonTree?
se it again"; 5 – "It was excellent")

How satisfied were you with the whole infrastructure (EU, LT, GH)?
(1 – "Never want to use it again"; 5 – "It was excellent")

Well-integrated toolchain

Grader Feedback

- Easier to check (due to automatic diagram renders)
- Faster to evaluate
  - No need to open the model
  - But sometimes rendering errors led to false results
- Quality of submissions generally higher
- Fewer issue tickets, technical problems
  - Despite the new toolchain!

Summary of Experiences

- ✓ **Fewer** technical problems
- ✓ Overall **positive** student experiences
- ✓ More **streamlined** grading

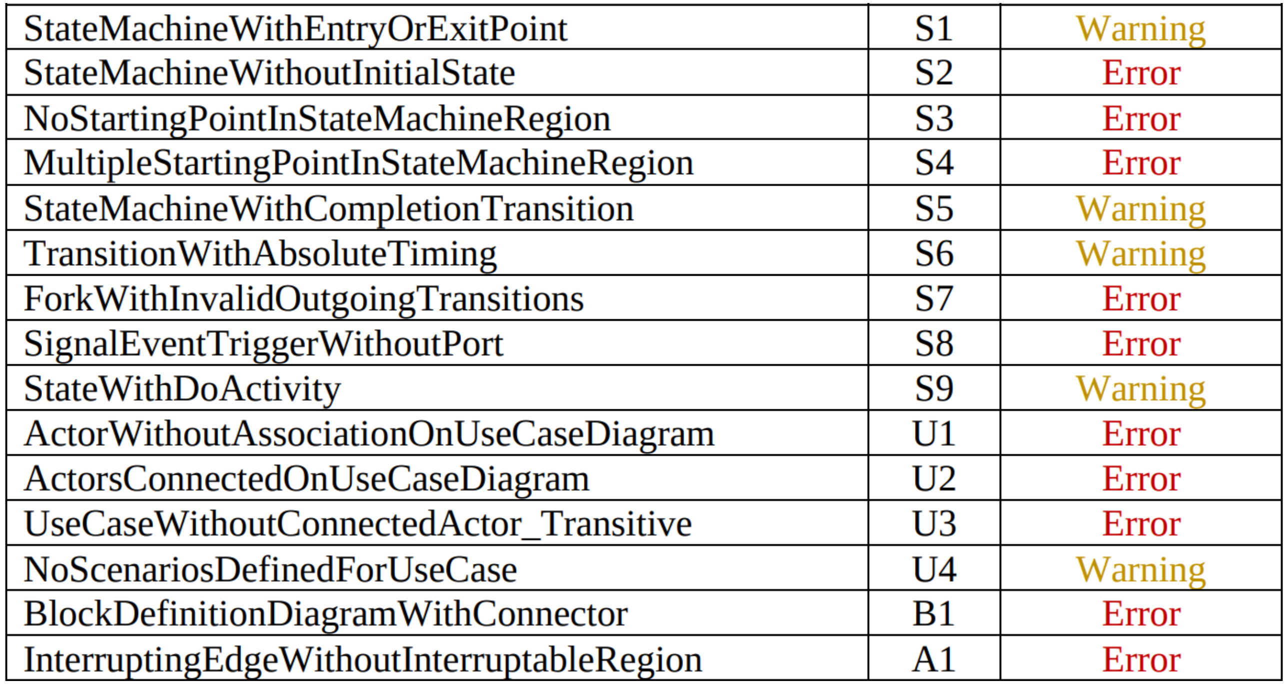
- ⓘ Modeling and model versioning are **hard**
- ⓘ More **training** for tools
- ⓘ Some rendering **errors** have to be investigated

Outlook: SysML v2

- New **challenges**
  - Education: new and complex language, new principles
  - Tooling: only preliminary tool support, requires deep knowledge
  - **Versioning:** textual syntax vs. graph model
- In its current state, the textual syntax is **only a view**
  - Need to be able to diff text **and** graphs
  - Display and interpret the diff as **graph models**
    - The *meaning* of changes is revealed in the graph
  - **Model slicing** is a very difficult problem in the context of merge
    - Different files might have fragments of the necessary information
    - Hard to preserve consistency of the model

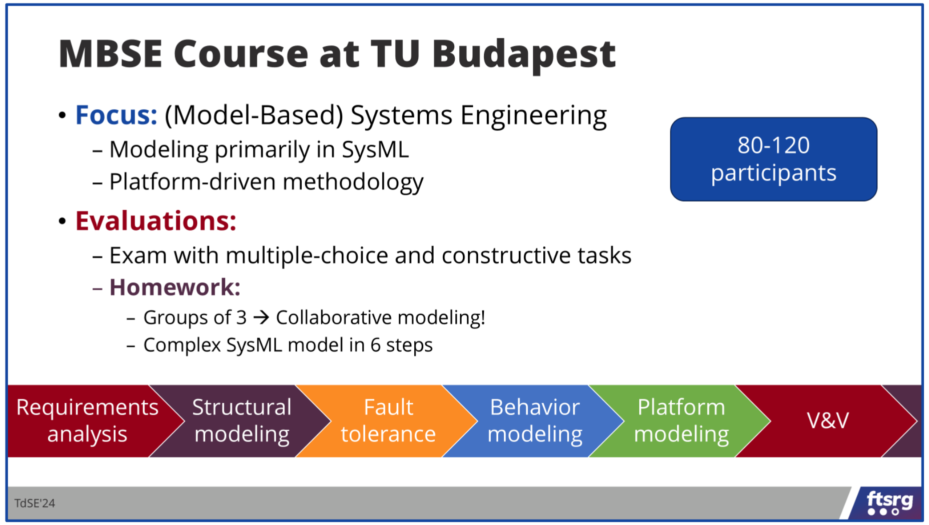
Future Plans

- Integrating a **pattern-based model validator**
  - Goal: semantic correctness of submissions
    - Not only structural requirements, but adhering to best practices as well
  - **Done**, testing in 2024/25/1 (Autumn) with limited patterns



StateMachineWithEntryOrExitPoint	S1	Warning
StateMachineWithoutInitialState	S2	Error
NoStartingPointInStateMachineRegion	S3	Error
MultipleStartingPointInStateMachineRegion	S4	Error
StateMachineWithCompletionTransition	S5	Warning
TransitionWithAbsoluteTiming	S6	Warning
ForkWithInvalidOutgoingTransitions	S7	Error
SignalEventTriggerWithoutPort	S8	Error
StateWithDoActivity	S9	Warning
ActorWithoutAssociationOnUseCaseDiagram	U1	Error
ActorsConnectedOnUseCaseDiagram	U2	Error
UseCaseWithoutConnectedActor_Transitive	U3	Error
NoScenariosDefinedForUseCase	U4	Warning
BlockDefinitionDiagramWithConnector	B1	Error
InterruptingEdgeWithoutInterruptableRegion	A1	Error

Summary



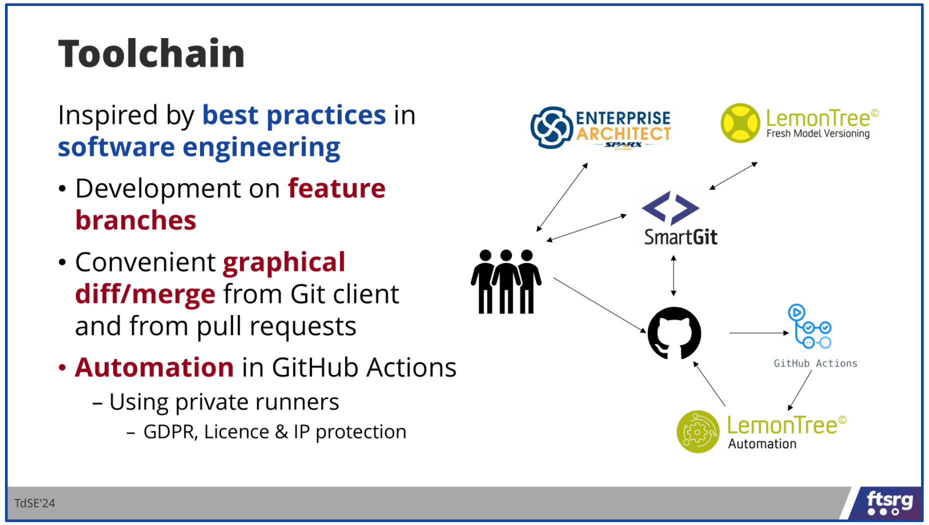
MBSE Course at TU Budapest

- **Focus:** (Model-Based) Systems Engineering
  - Modeling primarily in SysML
  - Platform-driven methodology
- **Evaluations:**
  - Exam with multiple-choice and constructive tasks
  - **Homework:**
    - Groups of 3 → Collaborative modeling!
    - Complex SysML model in 6 steps

80-120 participants

Requirements analysis
Structural modeling
Fault tolerance
Behavior modeling
Platform modeling
V&V

TdSE'24

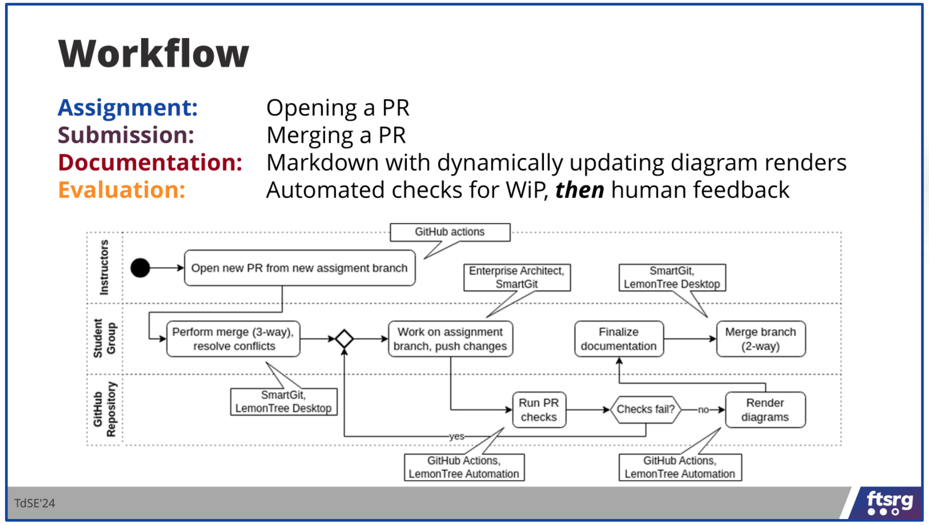


Toolchain

Inspired by **best practices** in **software engineering**

- Development on **feature branches**
- Convenient **graphical diff/merge** from Git client and from pull requests
- **Automation** in GitHub Actions
  - Using private runners
    - GDPR, Licence & IP protection

TdSE'24

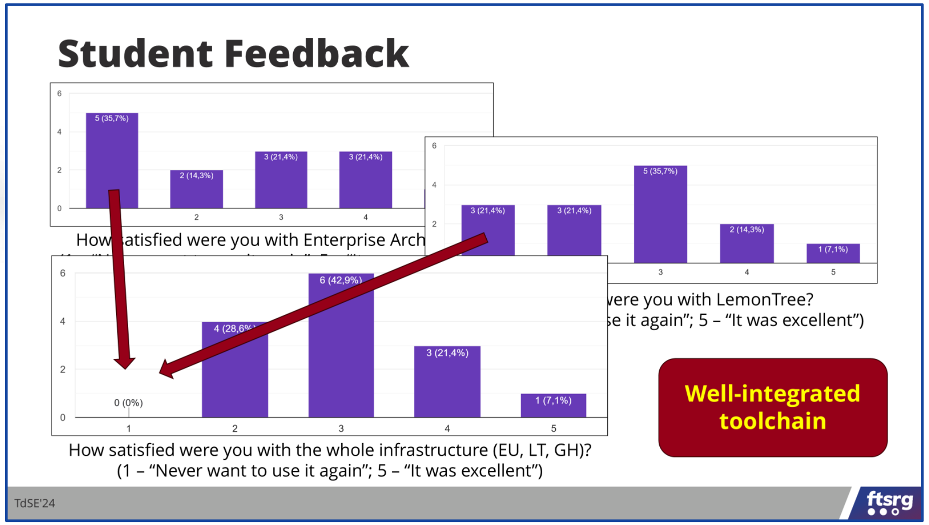


Workflow

Assignment: Opening a PR
Submission: Merging a PR
Documentation: Markdown with dynamically updating diagram renders
Evaluation: Automated checks for WiP, ***then*** human feedback

Instructors
Student Group
GitHub Repository
GitHub actions
Open new PR from new assigment branch
Enterprise Architect, SmartGit
SmartGit, LemonTree Desktop
Perform merge (3-way), resolve conflicts
Work on assignment branch, push changes
Finalize documentation
Merge branch (2-way)
SmartGit, LemonTree Desktop
Run PR checks
Checks fail?
no
Render diagrams
yes
GitHub Actions, LemonTree Automation
GitHub Actions, LemonTree Automation

TdSE'24



Student Feedback

How satisfied were you with Enterprise Arch

were you with LemonTree?
se it again"; 5 – "It was excellent")

How satisfied were you with the whole infrastructure (EU, LT, GH)?
(1 – "Never want to use it again"; 5 – "It was excellent")

Well-integrated toolchain

TdSE'24