Technical Debt in Cloud Native Geoserver Interview

Presentations :)

Are you familiar with the technical debt metaphor?

A brief introduction to technical debt

In software-intensive systems, technical debt is a collection of design or implementation constructs that are expedient in the short term, but set up a technical context that can make future changes more costly or impossible. Technical debt presents an actual or contingent liability whose impact is limited to internal system qualities, primarily maintainability and evolvability.

Our study

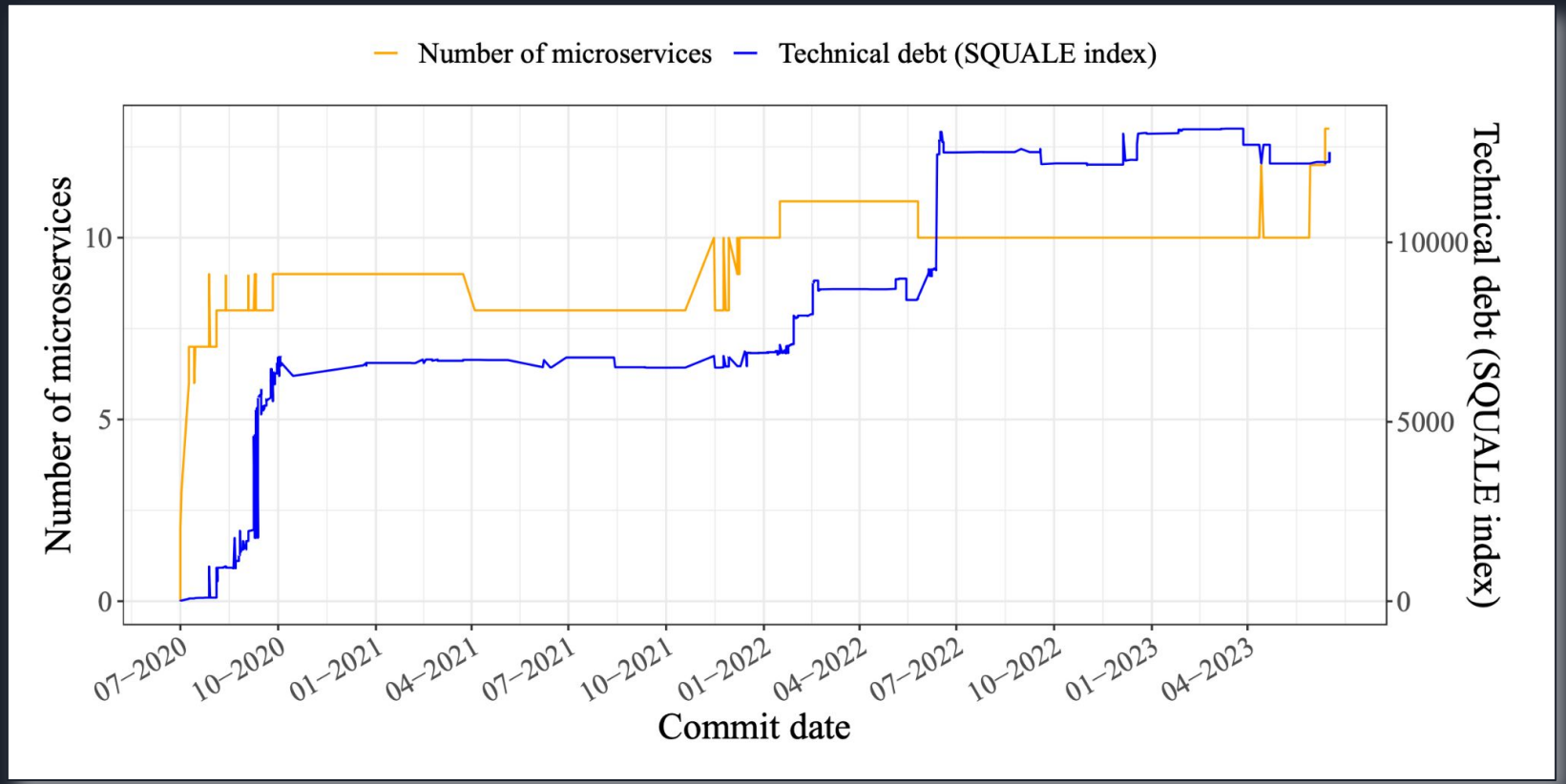
- Analyzed all Cloud Native Geoserver commits with SonarQube
- SonarQube automatically reviews code with static analysis to detect code smells
- Examples of code smells:
  - *Class variable fields should not have public accessibility*
  - *Locks should be released*
  - *Method overrides should not change contracts*
- *Each rule* is associated to a *technical debt refactoring time*
- *Technical debt* the time required to refactor all *code smells*

From the inspection of the Cloud Native Geoserver commits, we identified you as the leading developer of the repository. Is this assumption correct?

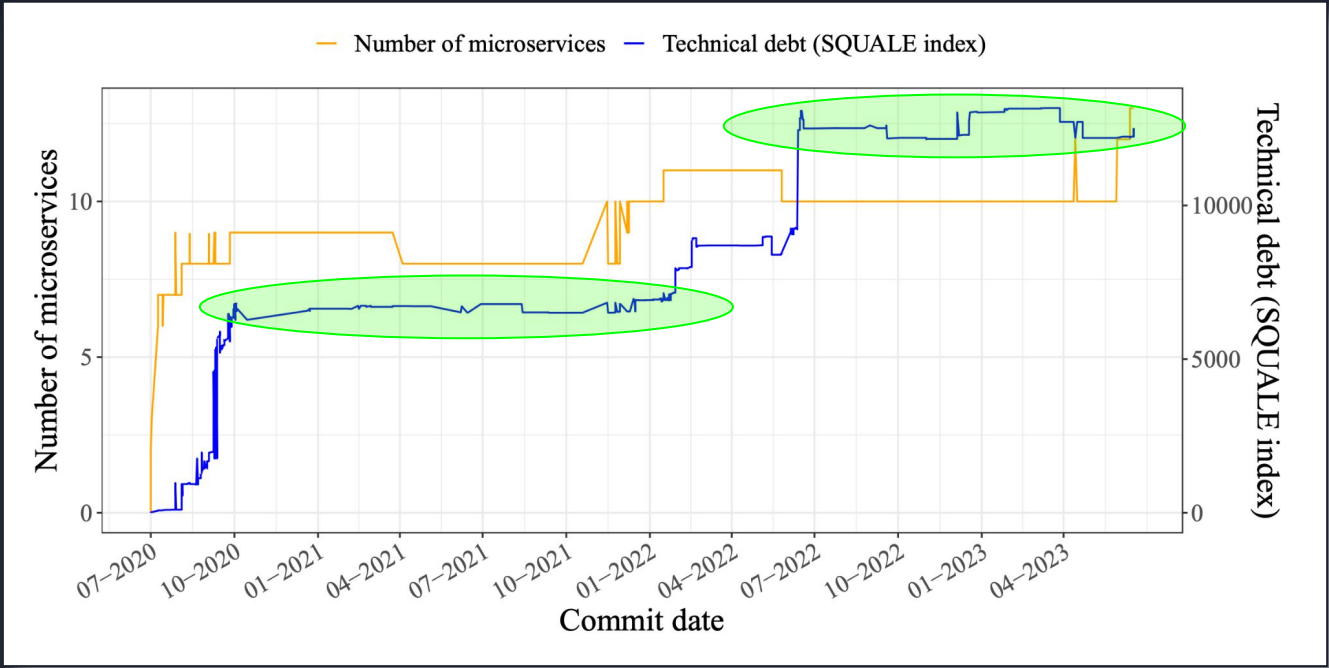
In a scale from 1 to 5, how familiar are you with the Cloud Native Geoserver?


Technical debt evolution
questions

Technical debt evolution



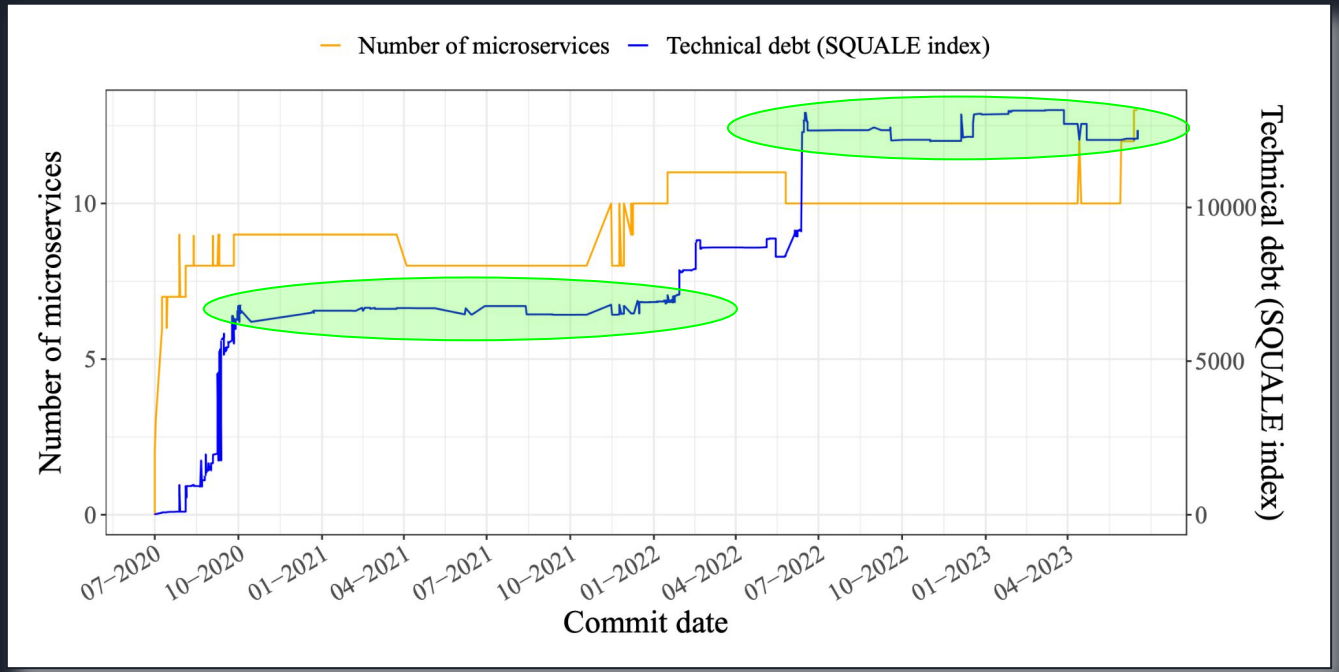
Question: Plateaus



We discovered long period of times when development continued but technical debt remained the same.

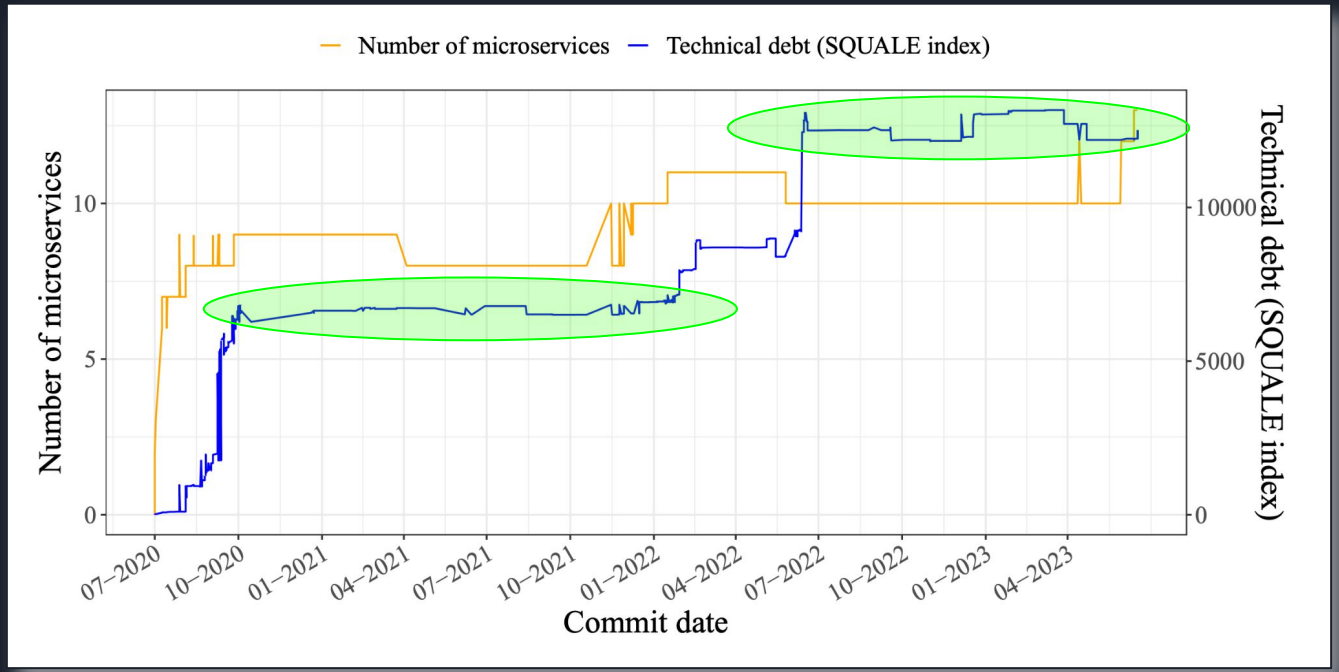
Could you provide us more information on this development periods?

Technical debt evolution



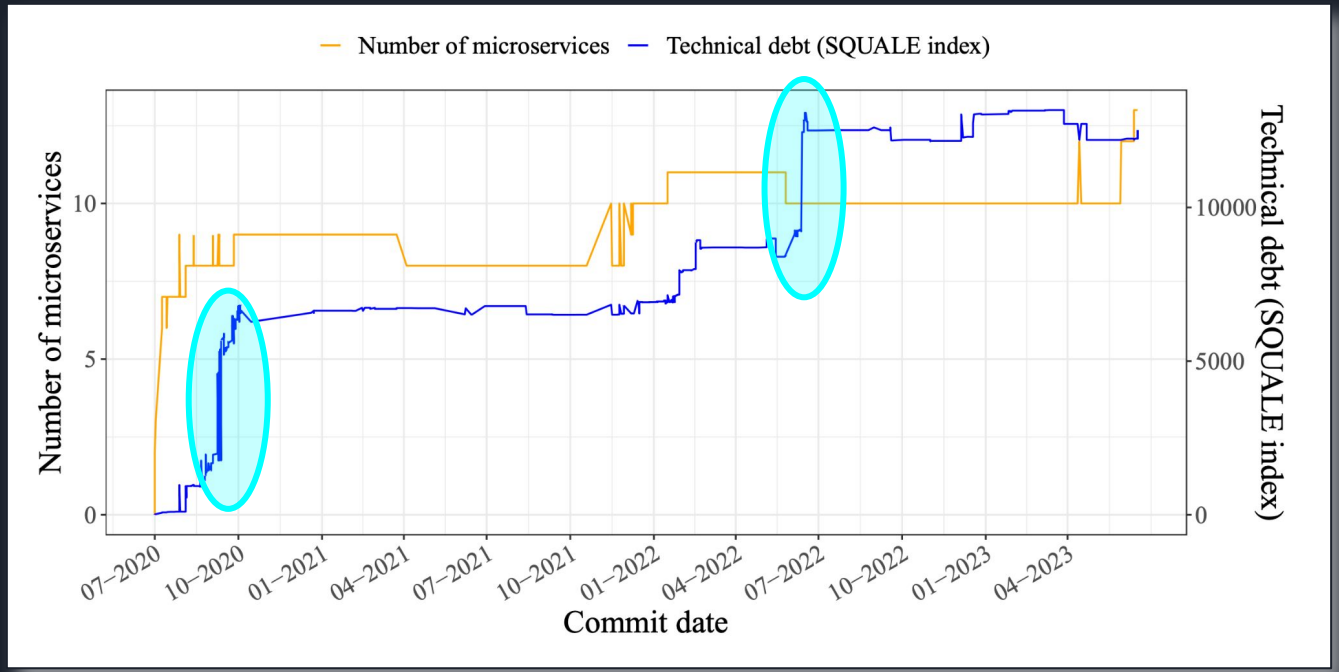
Do you recall which development activities did you conducted in this period?

Technical debt evolution

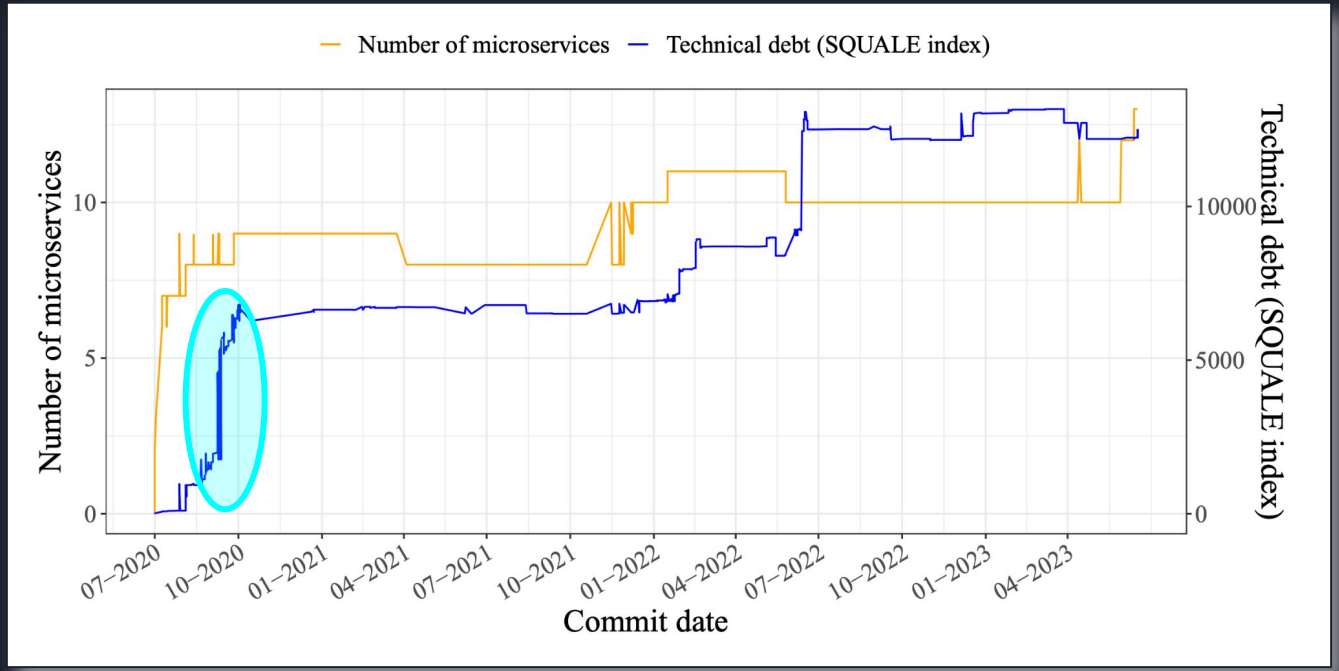


Did you purposely tried to avoid increasing the technical debt during this periods?

Question: Technical Debt Sudden Increases



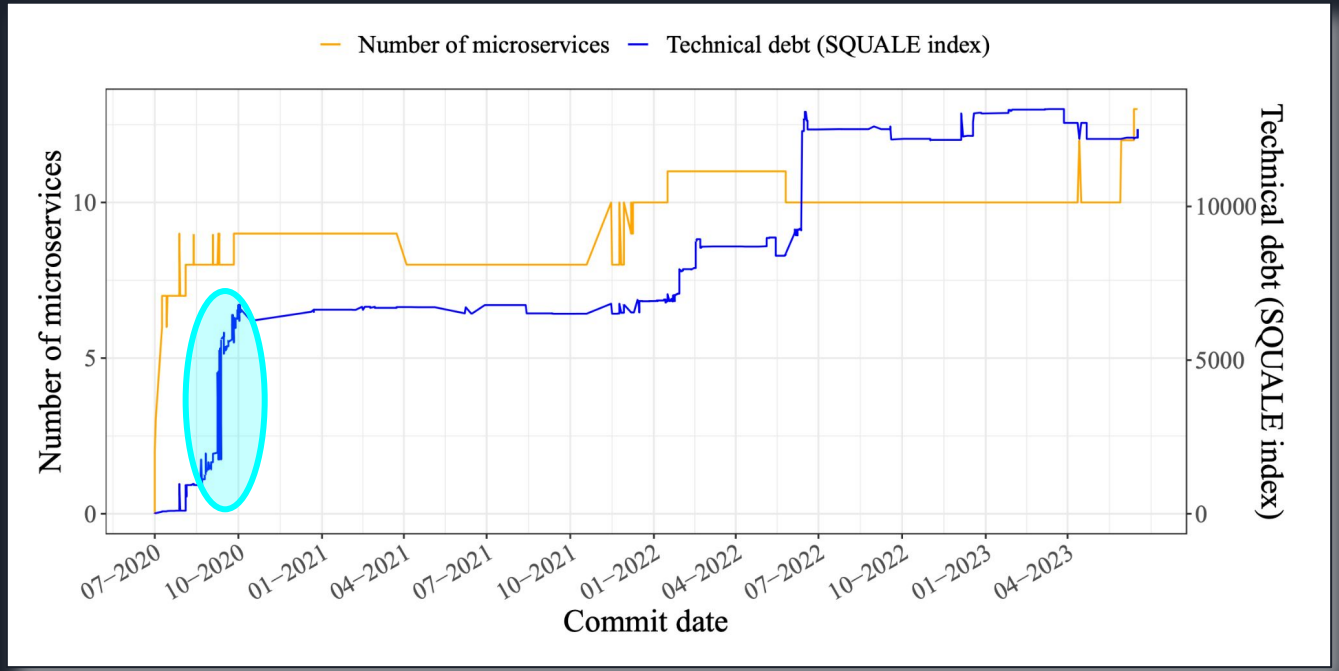
Question: Technical Debt Sudden Increases



TD “hotspot”: *Refactoring followed by a new binding to a JSON parser in a microservice*.

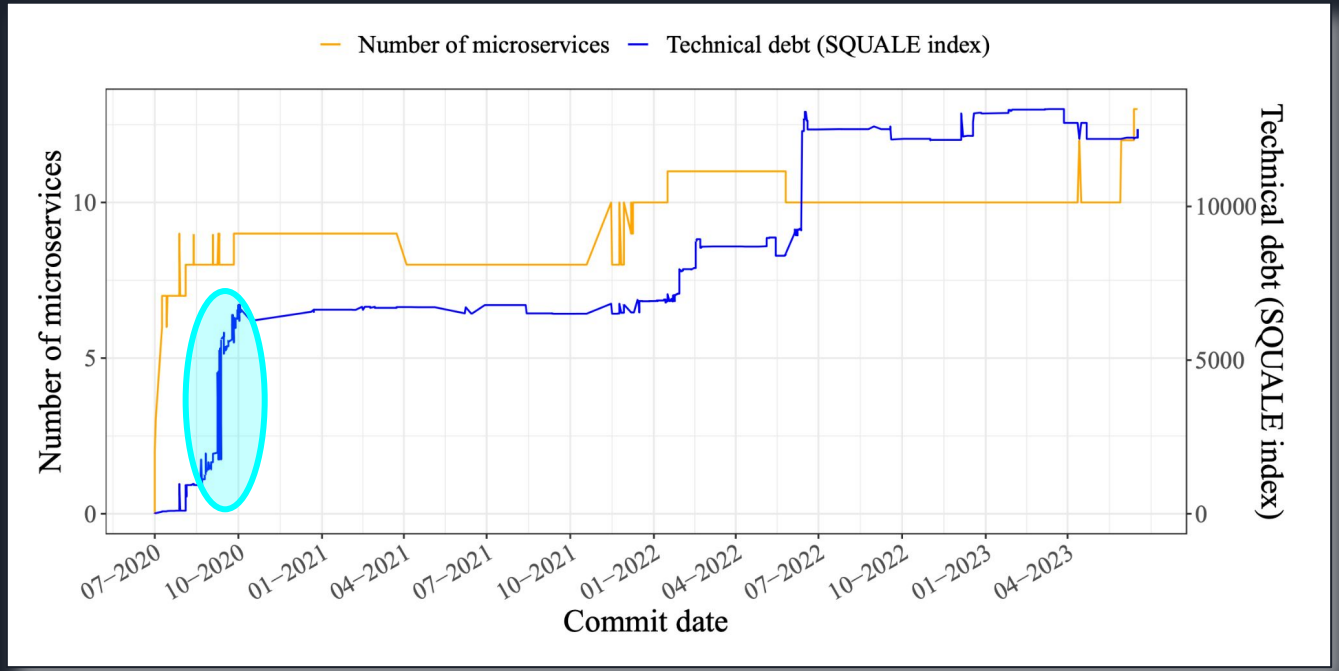
Could you provide us more information on this development period?

Question: Technical Debt Sudden Increases



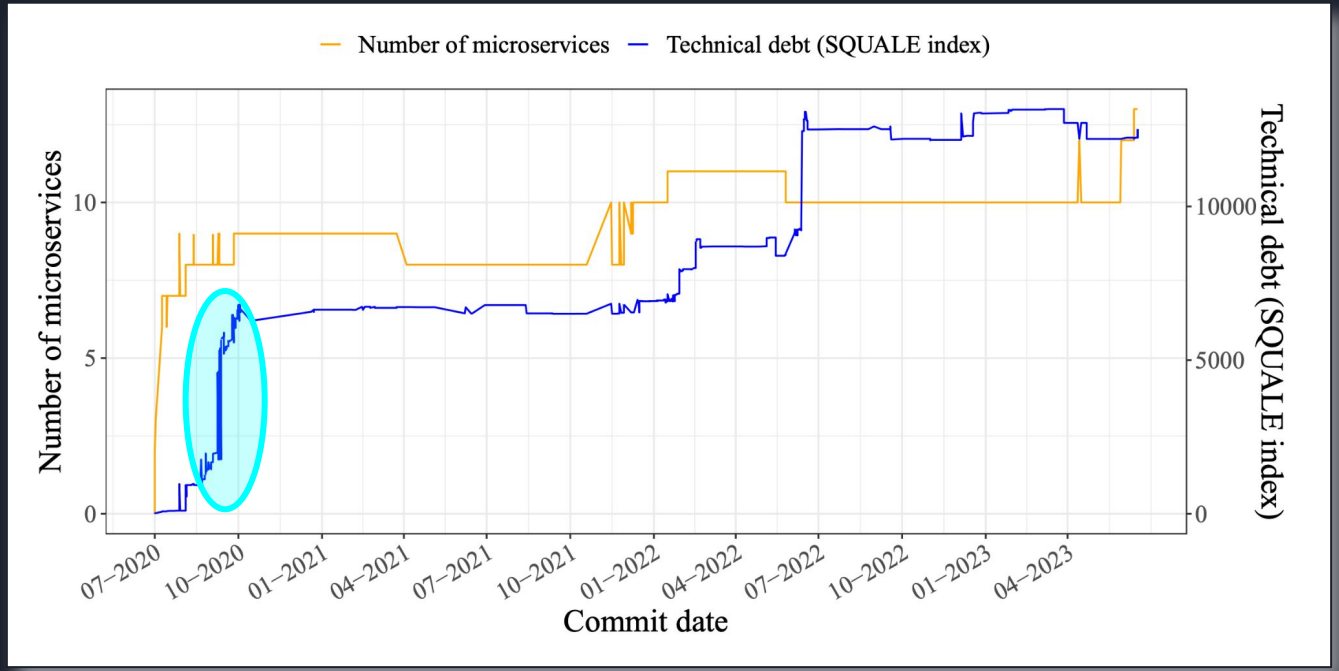
Do you recall which development activities did you conducted in this period?

Question: Increases



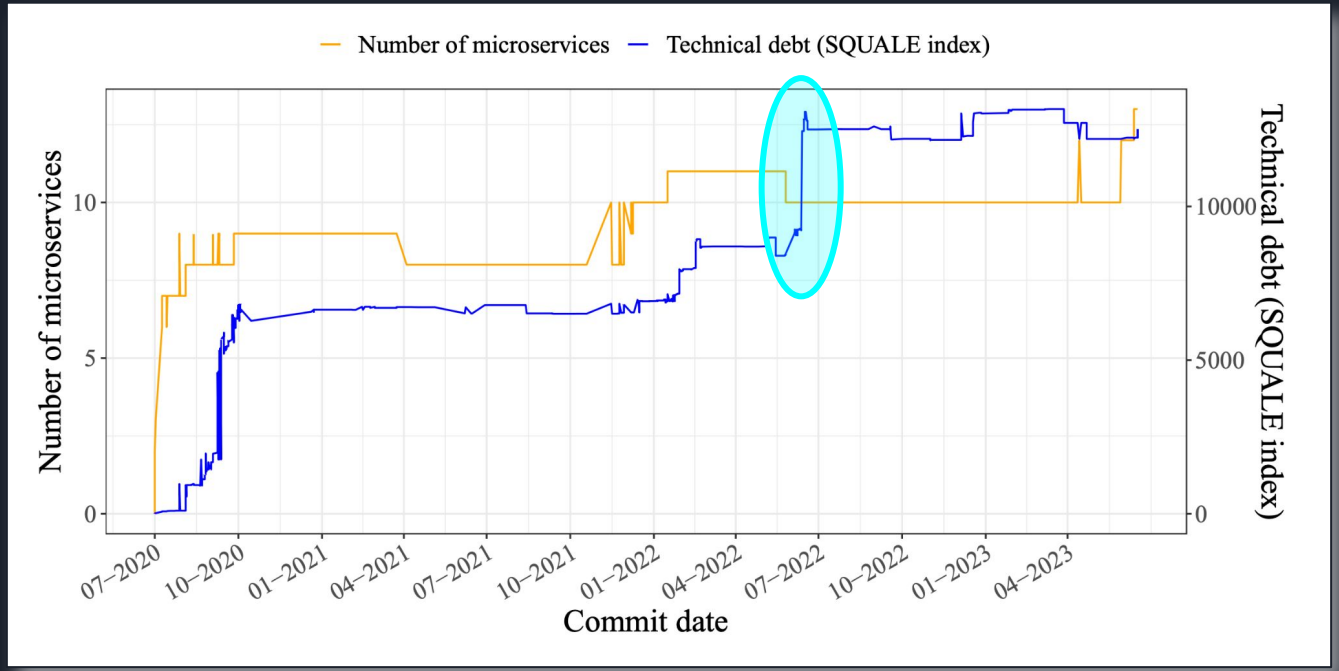
Where you aware of the technical debt you accumulated during this period?

Question: Increases



Did the technical debt increase affected future your development activities?

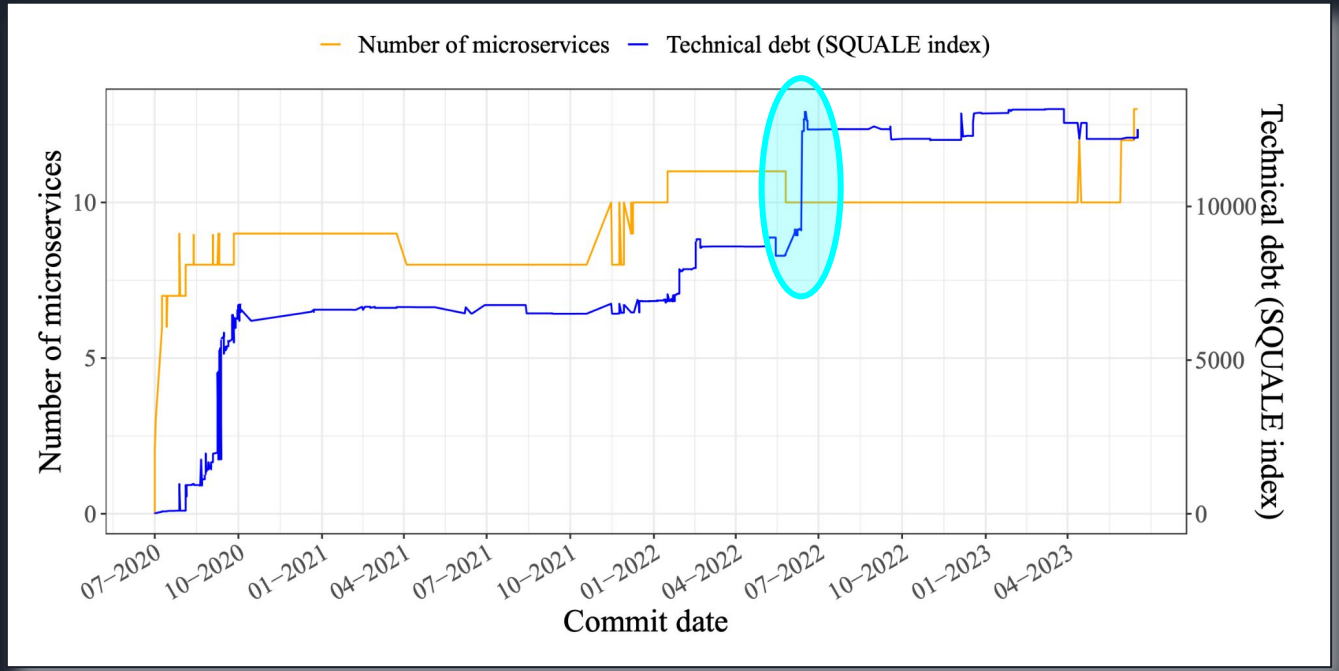
Question: Increases



TD “hotspot”: *JUnit upgrade*. Cross-microservice changes. Activities: Test refactoring, bug fixing, add logging, add automation

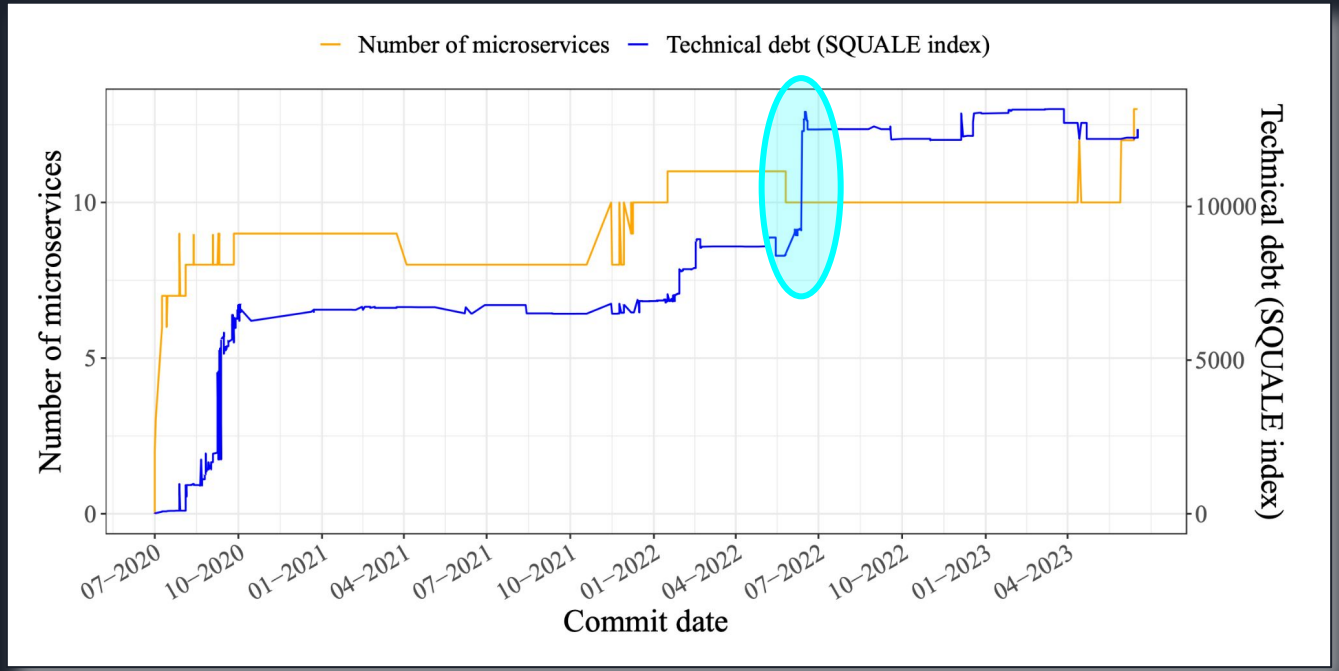
Could you provide us more information on this development period?

Question: Increases



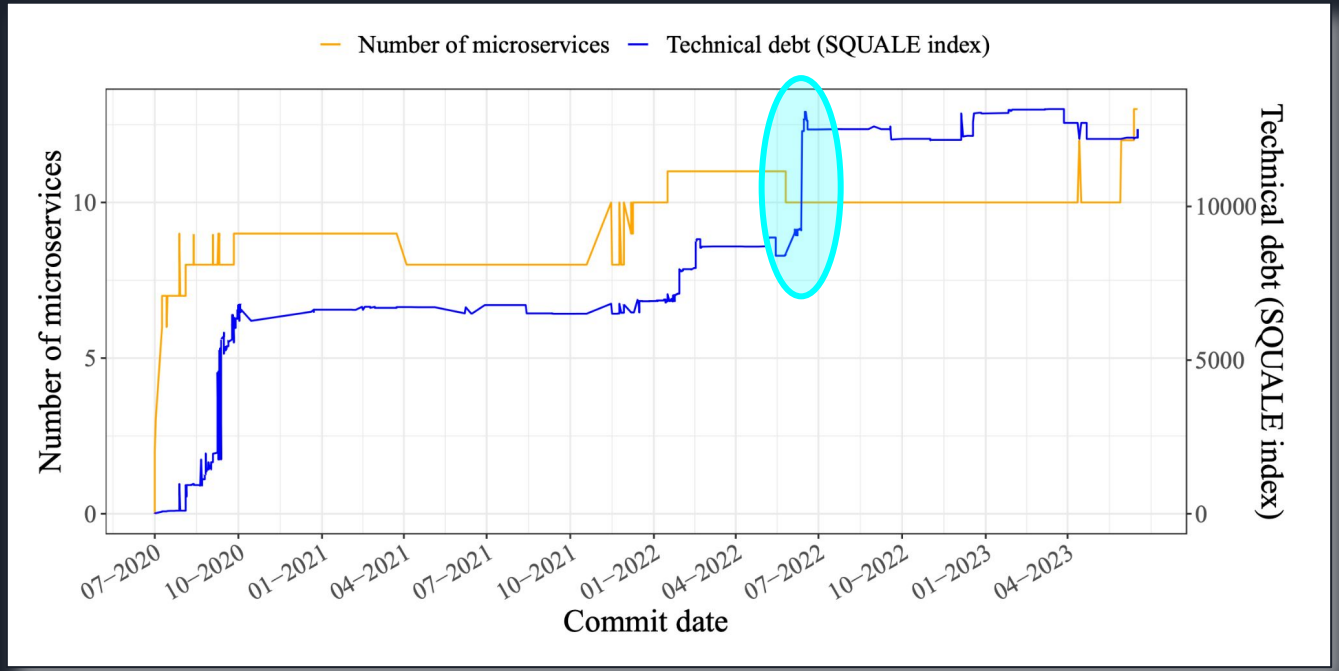
Do you recall which development activities did you conducted in this period?

Question: Increases

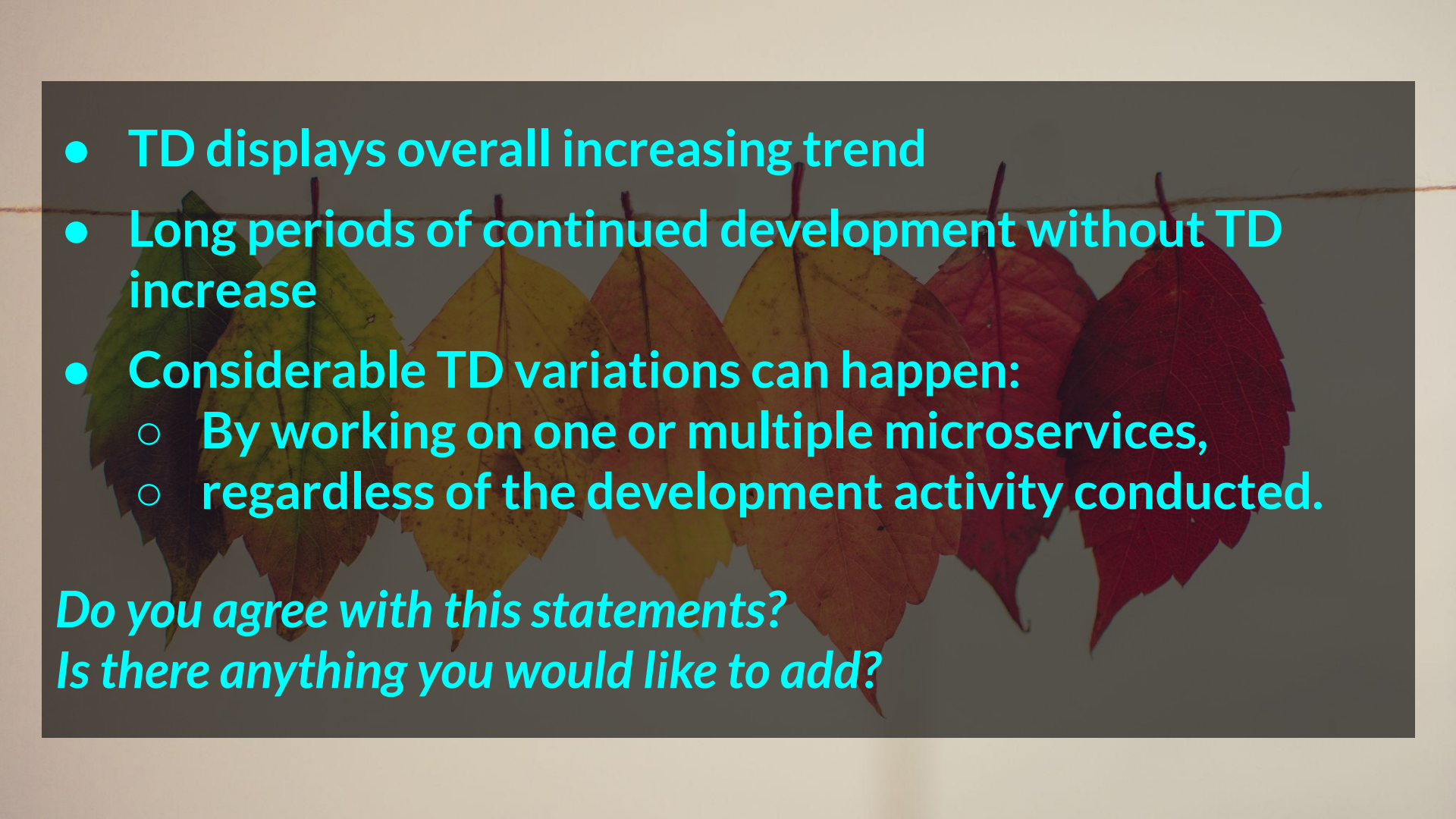


Where you aware of the technical debt you accumulated during this period?

Question: Increases

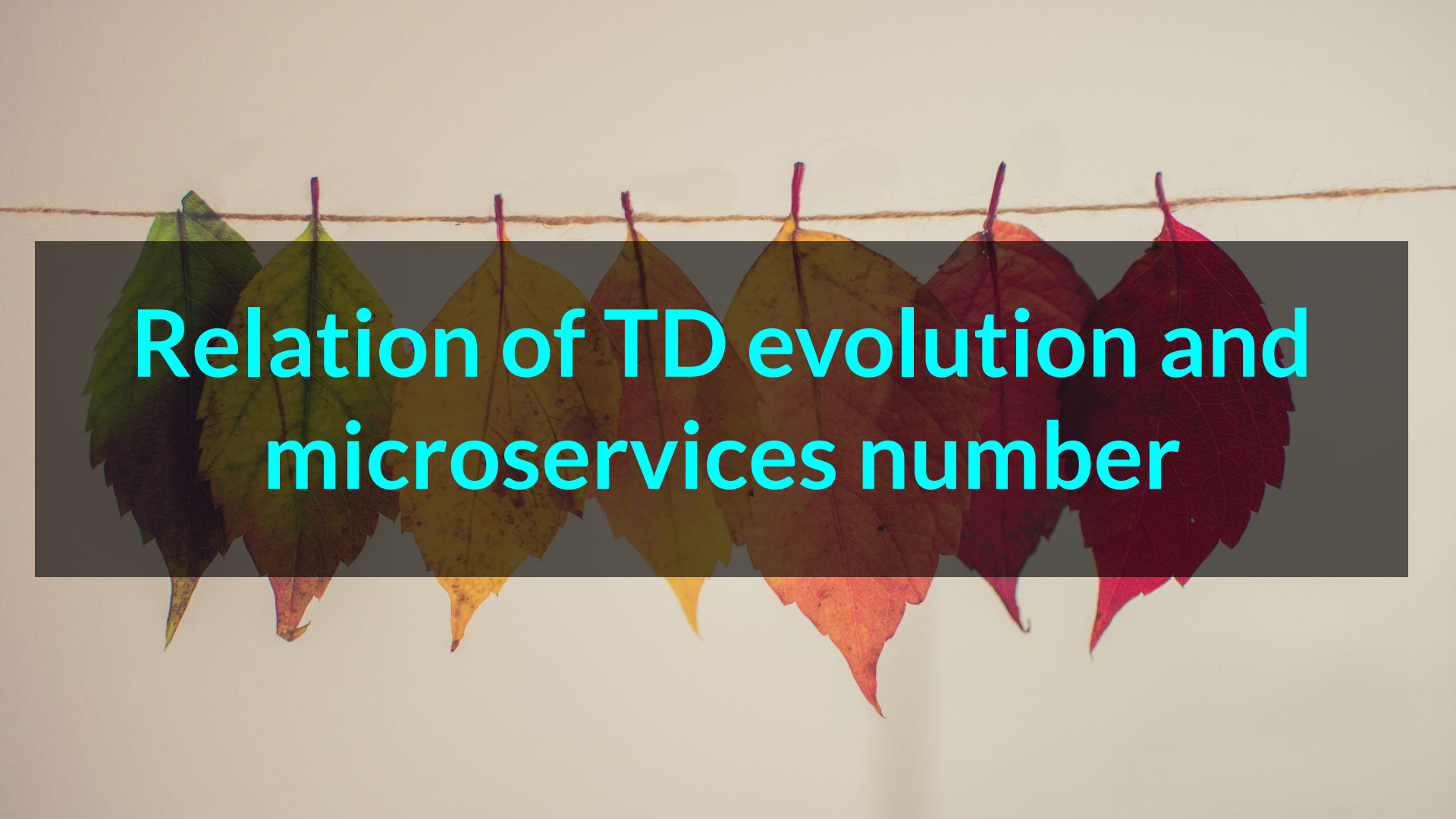


Did the technical debt increase affected future your development activities?

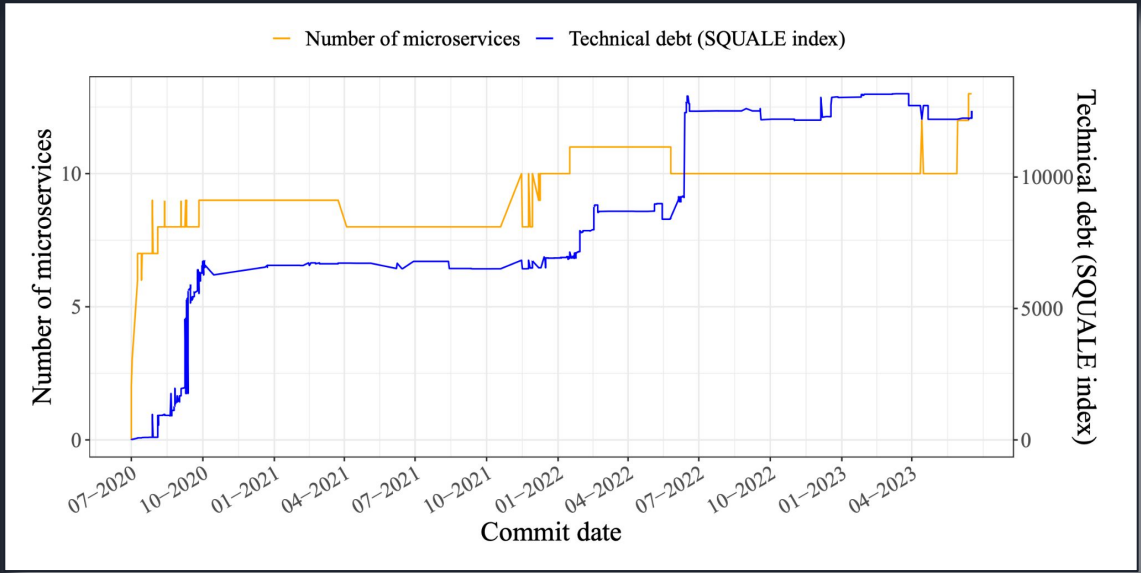


- TD displays overall increasing trend
- Long periods of continued development without TD increase
- Considerable TD variations can happen:
  - By working on one or multiple microservices,
  - regardless of the development activity conducted.

Do you agree with this statements?
Is there anything you would like to add?


Relation of TD evolution and microservices number

Relation of TD evolution and microservices number



- **Technical debt and number of microservices are strongly correlated**
- **Impact of adding / removing microservices** has **similar impact on TD regardless** of the **number of microservices present**
- Conjecture: *Correct **adherence to the microservice principles**, microservices are developed independently*

Question: Relation of TD evolution and microservices number

- **Technical debt and number of microservices are strongly correlated**
- **Impact of adding / removing microservices** has **similar impact on TD regardless** of the **number of microservices present**
- Conjecture: *Correct **adherence to the microservice principles**, microservices are developed independently*

Do you agree with this statements?

Could you provide us with more information on the relation of TD evolution and microservices number in Cloud Native Geoserver?

Is there anything you would like to add?

Is there anything else you want to add on technical debt in Cloud Native Geoserver?

Is there anything else you want to add on technical debt in microservice based architectures?