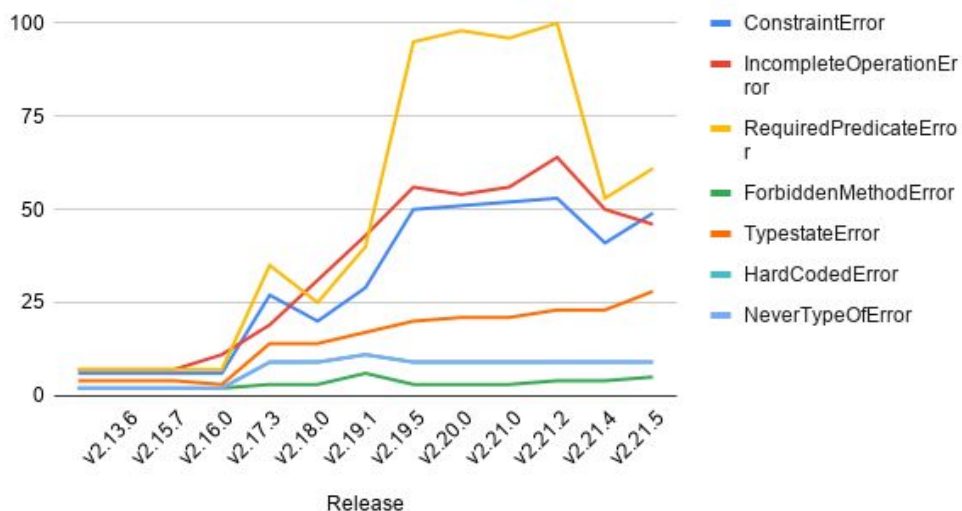


Jackrabbit

Possui 14 contribuidores expressivos (com mais de 10 commits) e por volta de 250 *releases*. Nas primeiras *releases* analisadas o número de erros é menor, mas conforme novas *releases* são lançadas ocorre um aumento no número de erros. Das classes de erros observadas, as que mais cresceram foram as de `RequiredPredicateError`, de `IncompleteOperationError` e `ConstraintError`. O gráfico de adições dá uma noção do tamanho do projeto.

- `ConstraintError`: média = 29 erros, variância = 395,6
- `IncompleteOperationError`: média = 43 erros, variância = 474,3
- `RequiredPredicateError`: média = 40 erros, variância = 1449,4
- `ForbiddenMethodError`: média = 3 erros, variância = 1,5
- `TypestateError`: média = 17 erros, variância = 75,5
- `HardCodedError`: média = 9 erros, variância = 12,3
- `NeverTypeOfError`: média = 9 erros, variância = 12,3

Análise de erros Jackrabbit

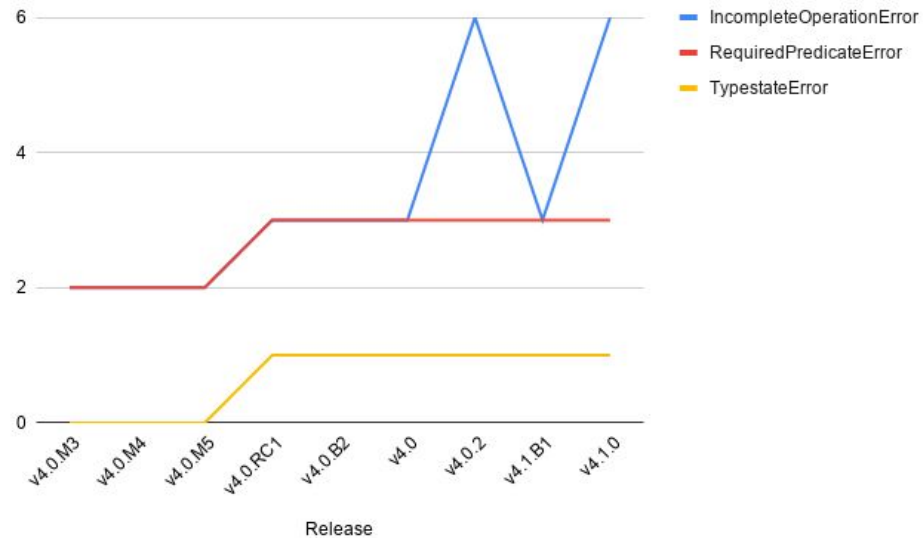


Cayenne

Possui por volta de 14 contribuidores expressivos e 44 releases. Ocorreu uma pequena variação no número de erros, mas no geral foi um número baixo.

- IncompleteOperationError: média = 3 erros, variância = 2,9
- RequiredPredicateError: média = 3 erros, variância = 0,2
- TypestateError: média = 1 erro, variância = 0,2

Análise de erros Cayenne

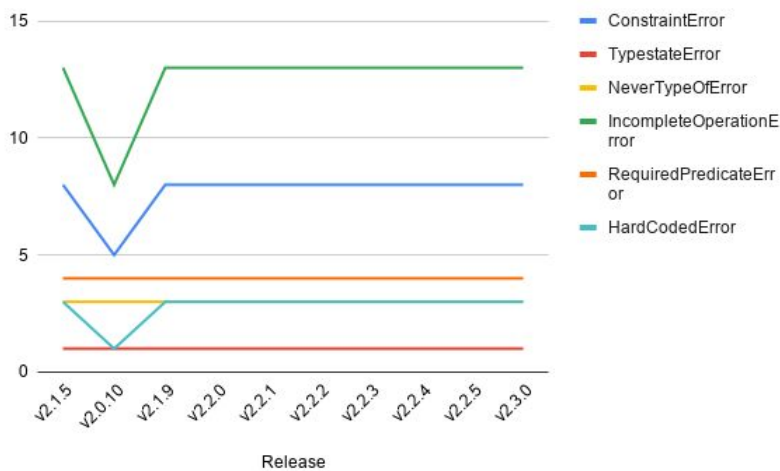


Wss4j

Possui por volta de 4 desenvolvedores expressivos e 250 releases. No geral, não ocorreu um aumento no número de erros, se mantendo constantes. Possivelmente, houveram correções ou apenas substituição de erros. Foram identificadas seis classes de erros:

- ConstraintError: média = 8 erros, variância = 0,9
- TypestateError: média = 1 erro, variância = 0.0
- NeverTypeOfError: média = 3 erros, variância = 0.0
- IncompleteOperationError: média = 13 erros, variância = 2,5
- RequiredPredicateError: média = 4 erros, variância = 0.0
- HardCodedError: média = 3 erros, variância = 0,4

Análise de erros Wss4j Security Commons

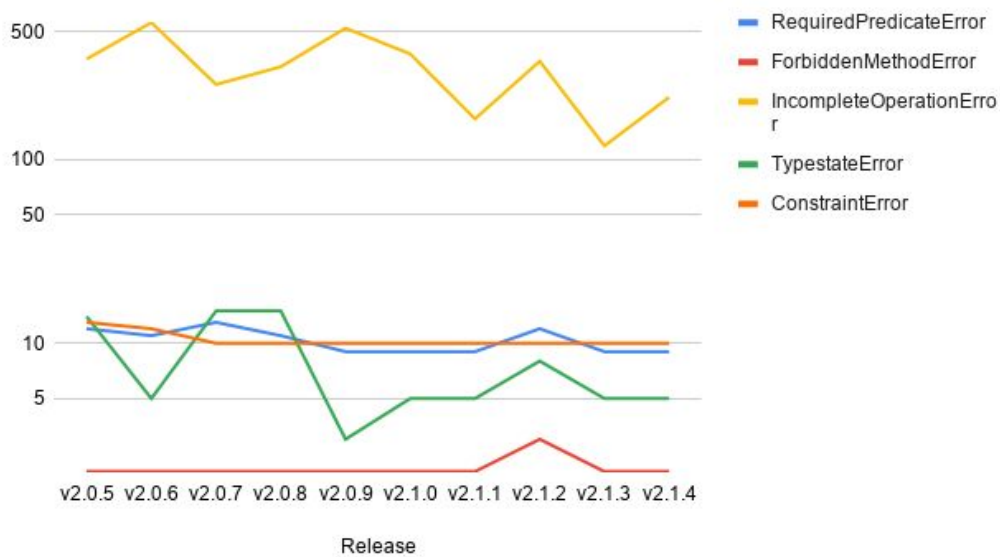


Santuário

Possui 2 desenvolvedores expressivos e 68 releases. Não foram observadas grandes variações ao longo das releases, com exceção de *IncompleteOperationError* que apresentou variações bruscas. Foram identificados cinco classes de erros:

- RequiredPredicateError: média = 10 erros, variância = 2,4
- ForbiddenMethodError: média = 2 erros, variância = 0,1
- IncompleteOperationError: média = 332,5 erros, variância = 20037
- TypestateError: média = 5 erros, variância = 22,6
- ConstraintError: média = 10 erros, variância = 0,1

Análise de erros Santuário

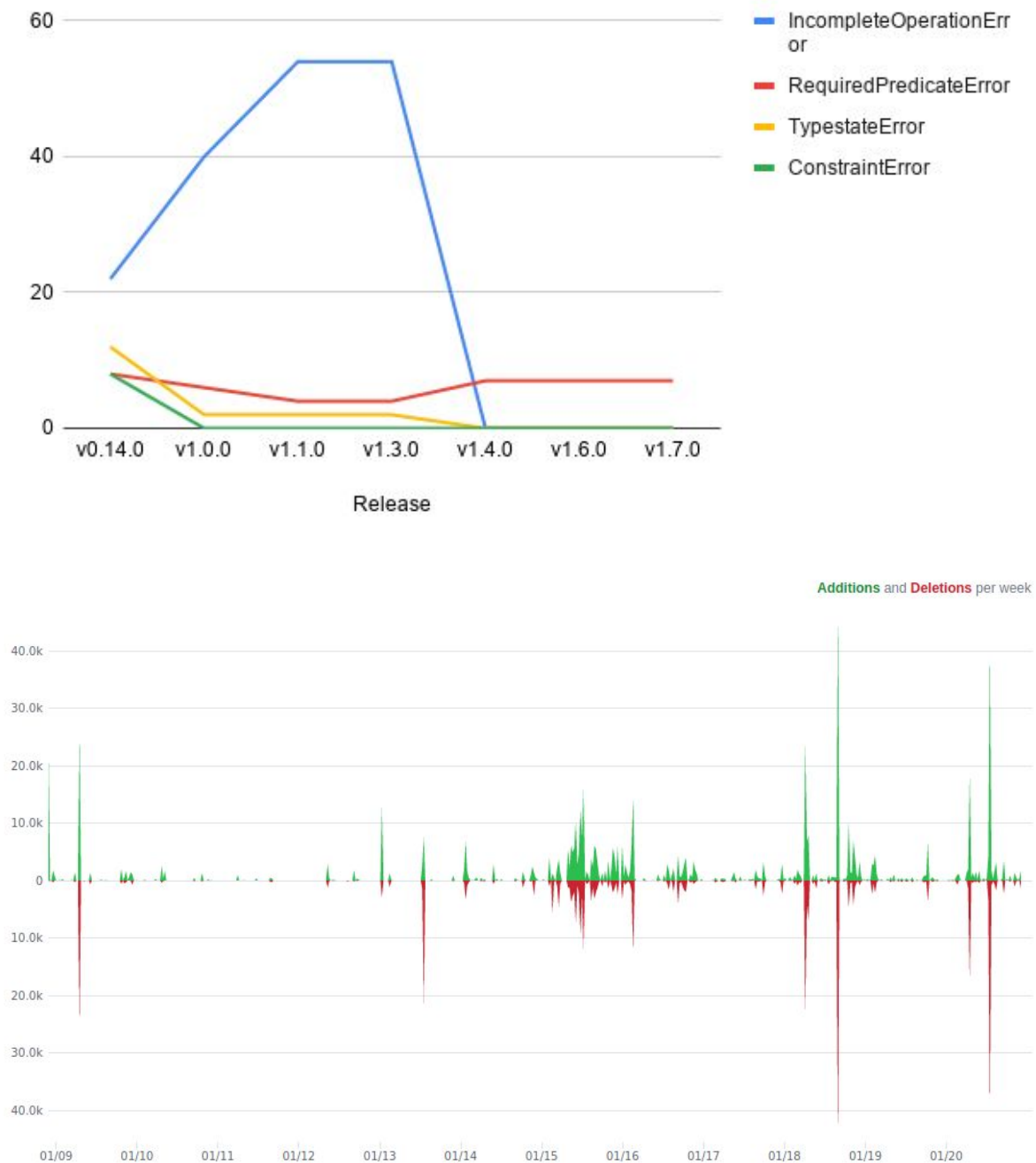


Mina

Possui 2 desenvolvedores expressivos e 32 releases. No geral, ocorreu uma diminuição no número de erros nas últimas releases, conforme mostra o gráfico. Foram identificadas as seguintes classes de erro:

- IncompleteOperationError: média = 22 erros, variância = 631
- RequiredPredicateError: média = 7 erros, variância = 2,7
- TypestateError: média = 2 erros, variância = 18,2
- ConstraintError: média = 0 erros, variância = 9,4

Análise de erros Mina



Conclusão

É possível observar que os erros em sua maioria ou se mantêm constantes ou sobem ao longo das *releases*. Em muitos casos isso se deve ao fato de que os projetos tendem a crescer e evoluir com novas *features* o que insere novos erros e não contribui para correção dos erros já observados. Apenas no caso do projeto Mina observou-se uma diminuição na quantidade de erros mas não se sabe se foi motivado por correções ou uma remoção de módulos defasados.

Nos casos do *Wss4j* e *Cayenne* os erros foram observados em módulos específicos que são núcleo do projeto e não sofreram alterações ao longo das versões. Nos projetos *Santuario* e *Jackrabbit* os erros já se apresentaram mais diluídos na aplicação e de difícil rastreamento, não se sabe se surgiram de novas bibliotecas adicionadas ou de implementações próprias do projeto. Pelo mesmo motivo é difícil saber o que pode ter causado a grande variação de ocorrência de um dos erros, como por exemplo *Incomplete Operation Error*. Este erro geralmente se apresenta associado a chamadas de funções que deveriam ter sido realizadas em um objeto porém não foram realizadas e o objeto já não mais se apresenta em memória.

Não foi possível observar uma correlação das quantidades de releases, desenvolvedores, tamanho de código nem nível de atividade dos projetos com os erros observados. A impossibilidade está fortemente atrelada à baixa quantidade de projetos analisados, que esteve limitada pela viabilidade de se compilar os arquivos fonte em tempo hábil. Alguns projetos possuem dependências não triviais de se resolver, como versões distintas de compiladores ou exigem muito tempo no processo algo em torno de mais de 1 hora sem contar com a análise do *Cognicrypt*.