lab-soft
software engineering laboratory

On the Use of FOP for Evolving Software Product Lines – A Comparative Study

Gabriel Coutinho Ferreira, Felipe Gaia,
Eduardo Figueiredo, Marcelo Maia

SBLP -> Science of Computer Programming

Presented by Gustavo Vale

Software Engineering Lab (LabSoft)
http://labsoft.dcc.ufmg.br/

Introduction

- The potential benefits of SPLs are achieved through a software architecture designed to increase reuse of features in several SPL products
  - Variability management is a key factor to be considered when evolving SPLs
    - Approaches annotative and compositional support variability management
  - SPL architecture stability and facilitate future changes

Variability mechanisms

- Should support
  - Non-intrusive
  - Self-contained changes that favor insertions
  - Not require deep modifications in existent components

Their Goal

- Find out how variability mechanisms behave in terms of modularity and change propagation on specific SPL change scenarios
  - Two case studies (SPLs) that evaluates comparatively three mechanisms
    - Conditional Compilation (CC)
    - Object-Oriented Design Patterns (DP)
    - Feature-Oriented Programming (FOP)

The evolved WebStore SPL

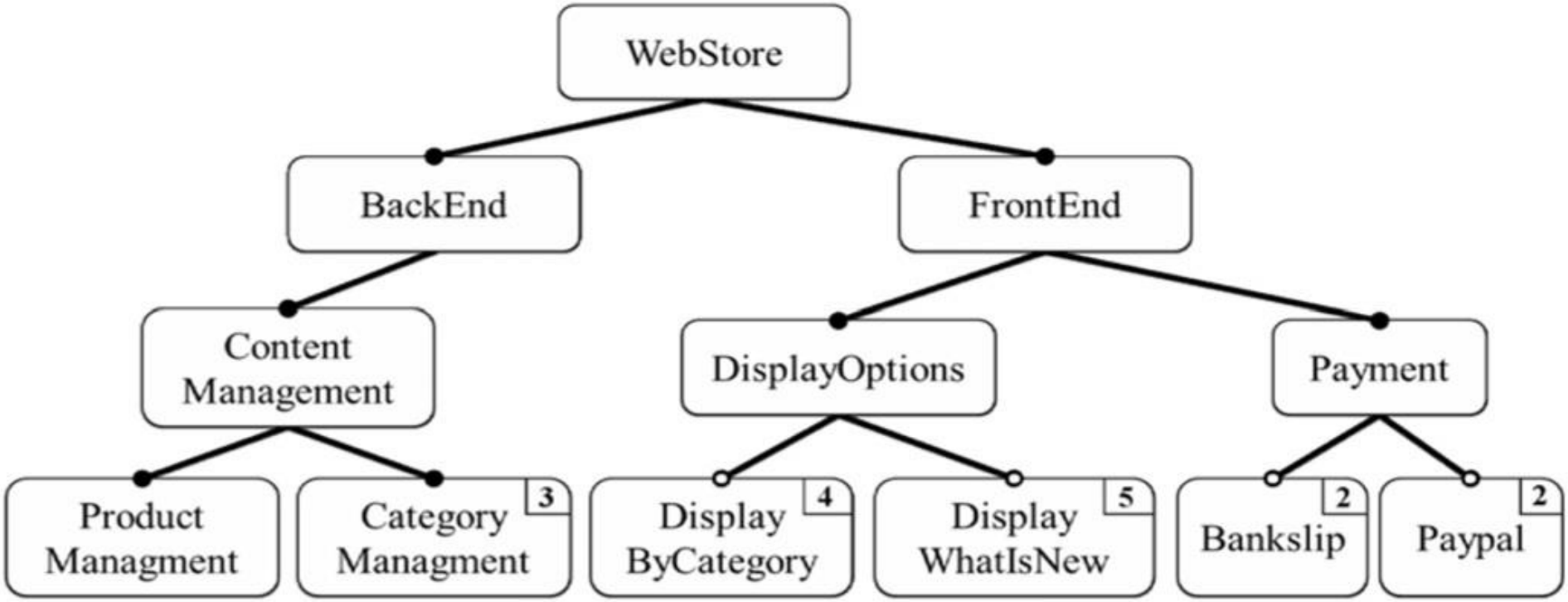


Fig. 1. WebStore basic feature model.

Summary of Scenarios in WebStore

Release	Description	Type of change
R1	WebStore core	
R2	Two types of payment included (Pay-pal and BankSlip)	Inclusion of optional feature
R3	New feature included to manage category	Inclusion of optional feature
R4	The management of category was changed to mandatory feature and new feature included to display products by category	Changing optional feature to mandatory and inclusion of optional feature
R5	New feature included to display products by nearest day of inclusion	Inclusion of optional feature

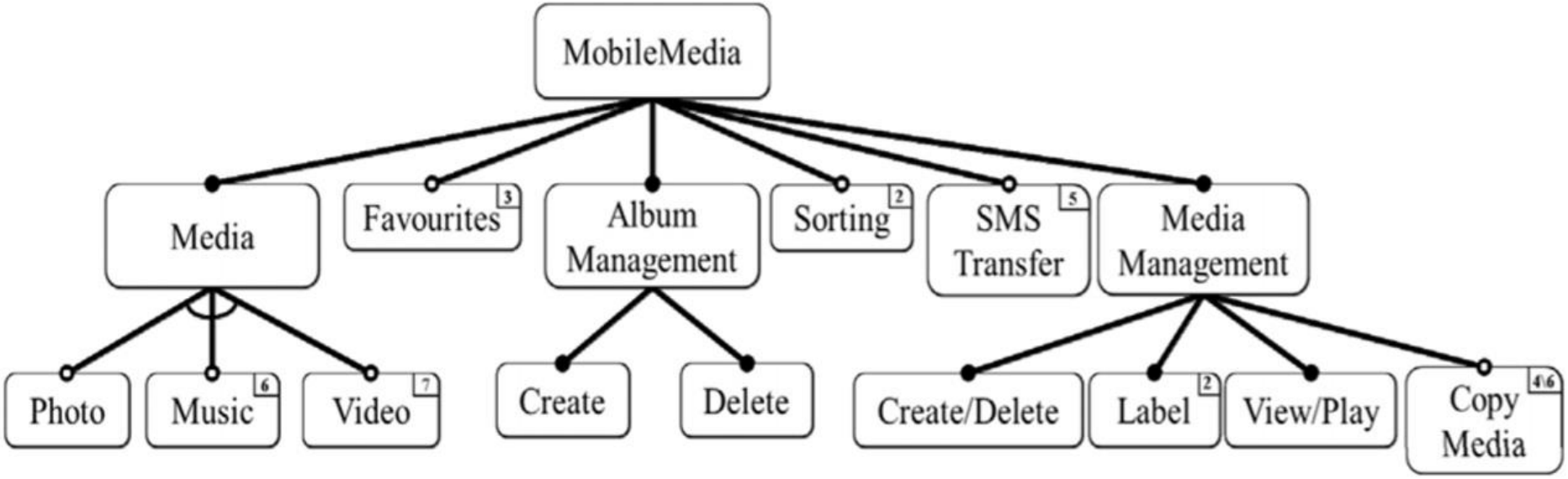
WebStore SPL Measures

	CC				
	R.1	R.2	R.3	R.4	R.5
#Components	23	23	26	26	26
#Methods	138	139	165	164	167
LOC (approx.)	885	900	1045	1052	1066

	FOP				
	R.1	R.2	R.3	R.4	R.5
#Components	25	35	44	41	47
#Methods	150	170	200	198	208
LOC (approx.)	945	1077	1257	1244	1303

	DP				
	R.1	R.2	R.3	R.4	R.5
#Components	28	32	38	40	44
#Methods	142	147	175	177	182
LOC (approx.)	915	950	1107	1121	1149

The evolved MobileMedia SPL



Scenarios in MobileMedia

Release	Description	Type of change
R1	MobileMedia core.	
R2	New feature added to count the number of times a photo has been viewed and sorting photos by highest viewing frequency. New feature added to edit the photo's label.	Inclusion of optional and mandatory features
R3	New feature added to allow users to specify and view their favorite photos.	Inclusion of optional feature
R4	New feature added to allow users to keep multiple copies of photos.	Inclusion of optional feature
R5	New feature added to send photo to other users by SMS.	Inclusion of optional feature
R6	New feature added to play music. The photo management basic features were generalized to manage media and ViewPhoto was turned into an alternative feature.	Changing of one mandatory feature into two alternatives
R7	New feature added to manage videos	Inclusion of alternative feature

Study Settings

- Research Questions
  - RQ1 – Does the use of FOP has smoother change propagation impact that CC and DP during the evolution of an SPL?
  - RQ2 – Does the use of FOP provides more modular and stable design than CC and DP of the SPL features in evolution?

Infrastructure Setting

- Independent Variable
  - Variability Mechanism
- Dependent Variables
  - Change Propagation Measures
  - Modularity Metrics

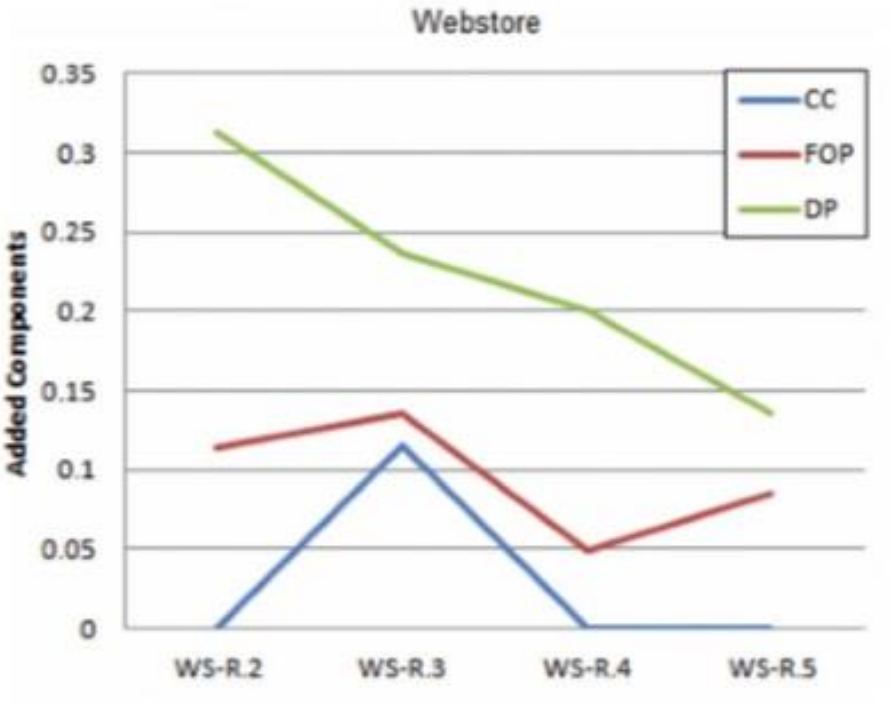
Study Phases

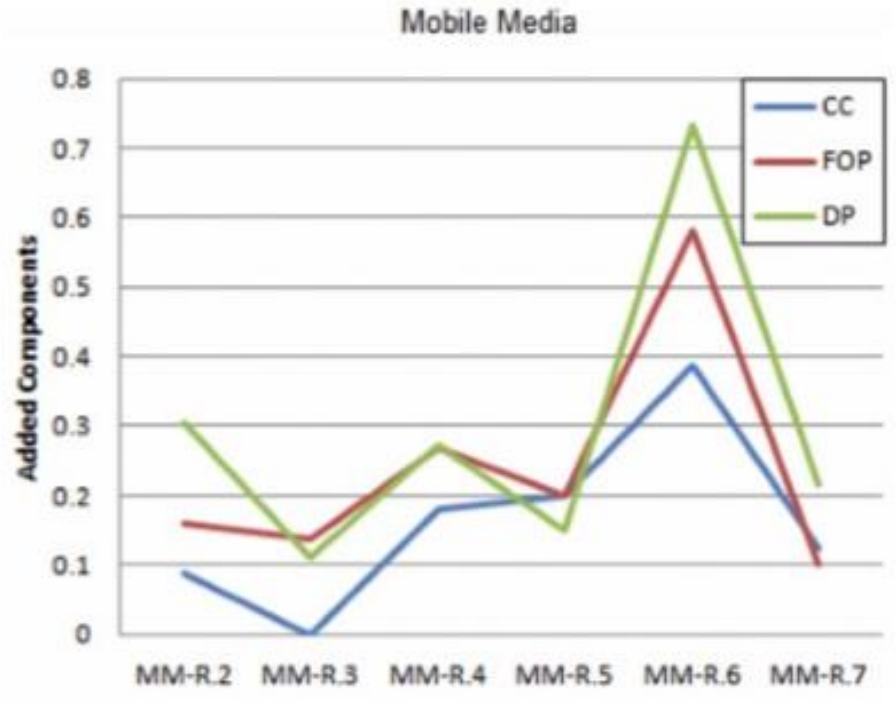
- Four phases
  - Construction of two subjects SPLs with complete releases
  - Manual feature assignment of all produced source code
  - Change propagation measurement and modularity metrics calculation
  - Quantitative and qualitative analysis of the results

Change Propagation Analysis

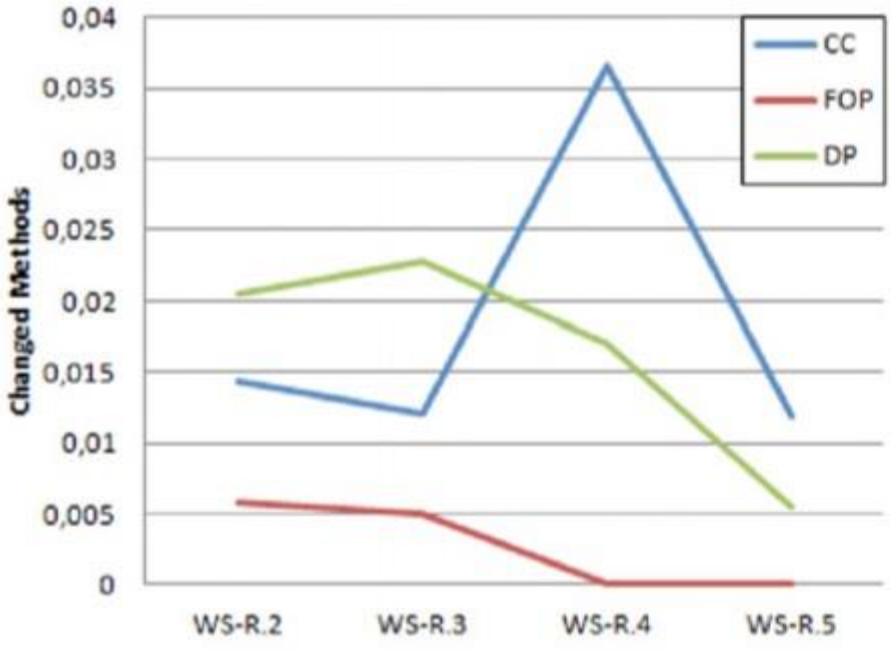
- Quantitative Analysis
  - Traditional measures
  - Granularity levels
    - Components, Methods and Lines of Source Code
      - Added, Removed and Changed
  - Lower number of modified and removed
    - Stable solutions
  - Lower number of added
    - Is not being supported by non-intrusive extensions

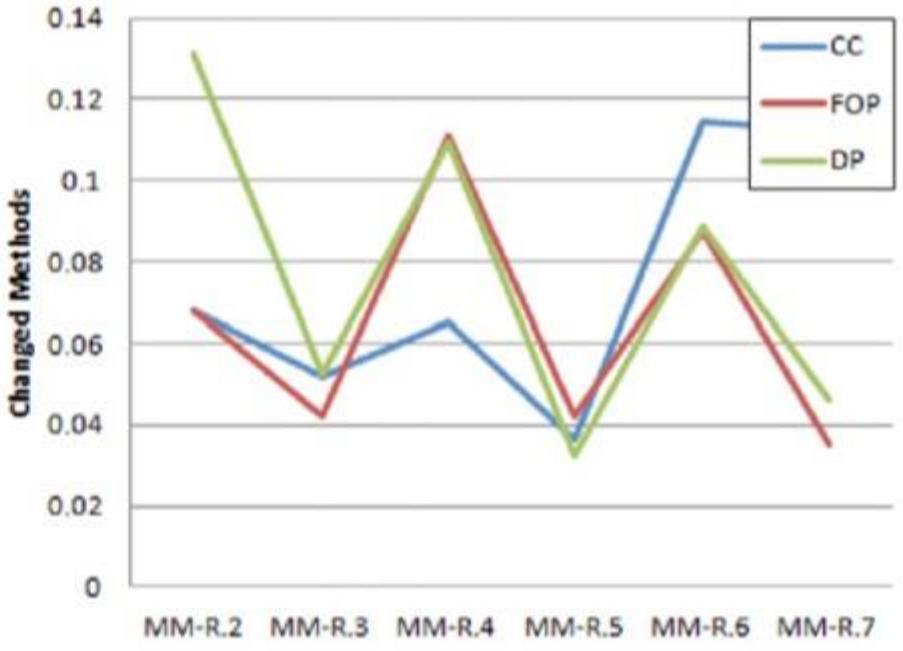
Added Components



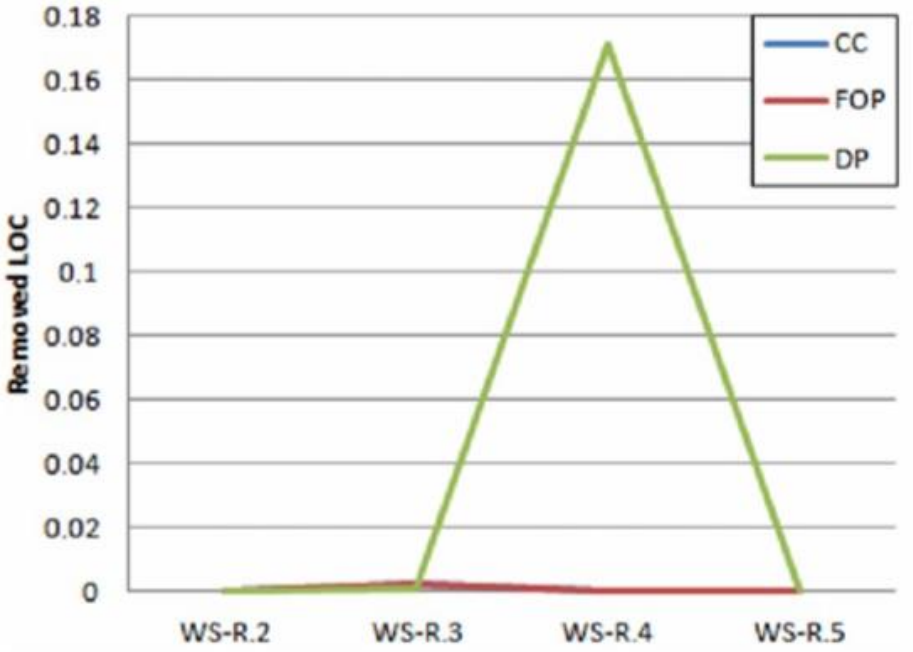


Changed Methods





Removed Lines of Code



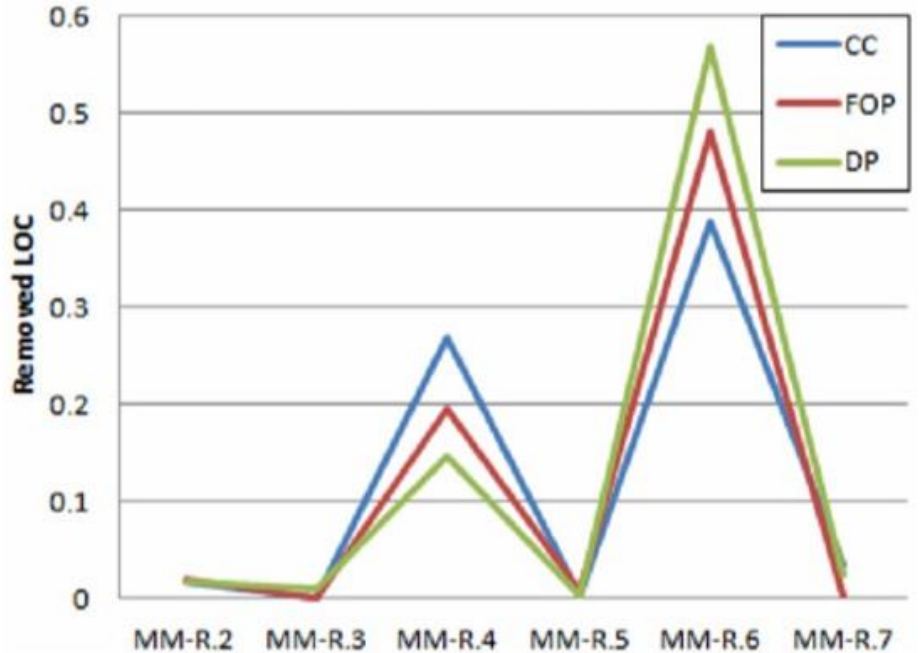


Fig. 5. Removals in WebStore and MobileMedia.

Change of Propagation Discussion

- CC releases have lower number of added components
- Not observed a significant difference between FOP and DP mechanisms
  - WebStore
    - DP introduces more components
  - MobileMedia
    - FOP the inverse situation in three of four change scenarios

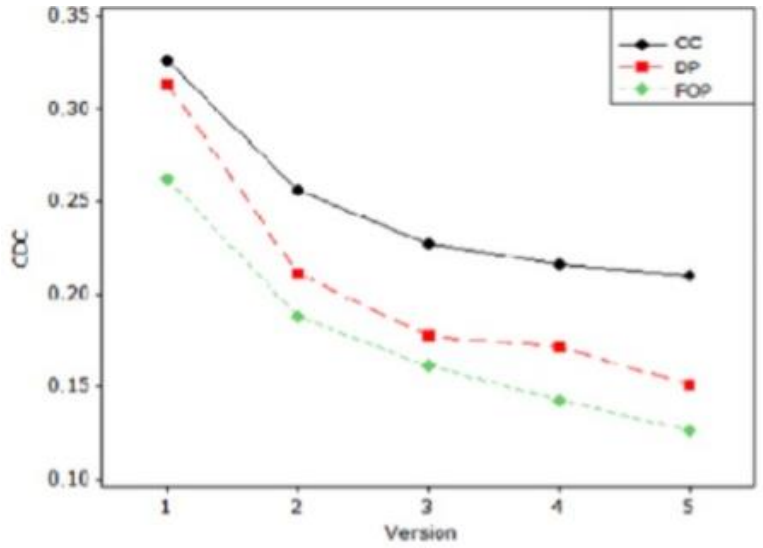
RQ1

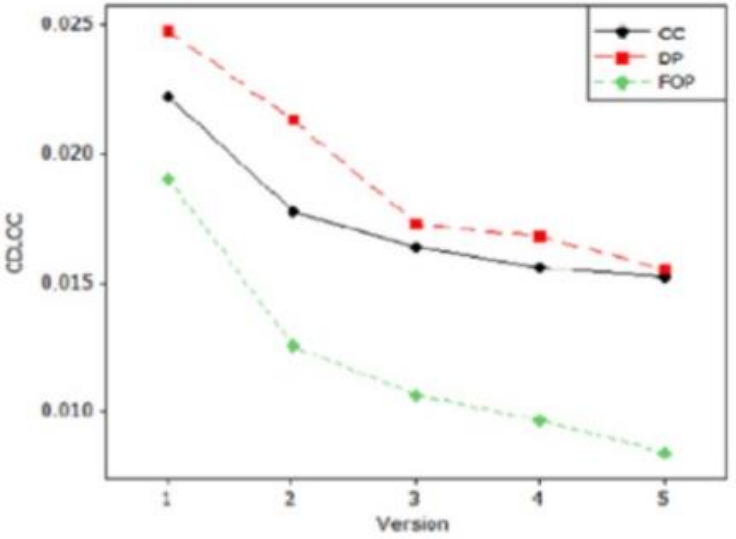
- Does the use of FOP has smoother change propagation impact that CC and DP during the evolution of an SPL?
  - FOP is slightly better

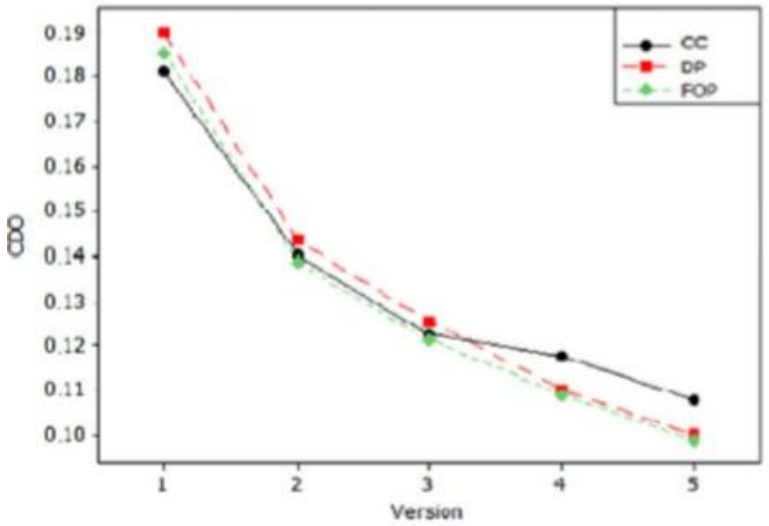
Modularity Analysis

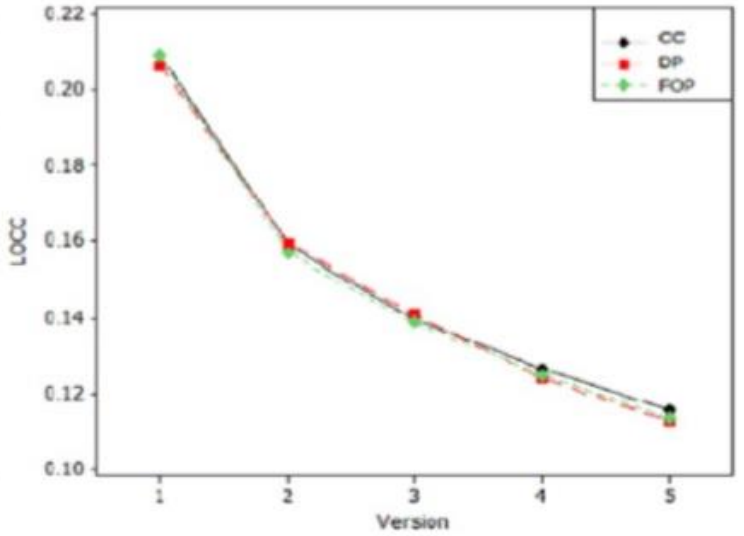
- Used a suite of metrics
  - Components – based on CDC
  - Operations (i.e. methods) – based on CDO
  - Lines of Code – based on CDLOC and LOCC
- For all this metrics, a lower values implies a better result

Metrics Values – WebStore

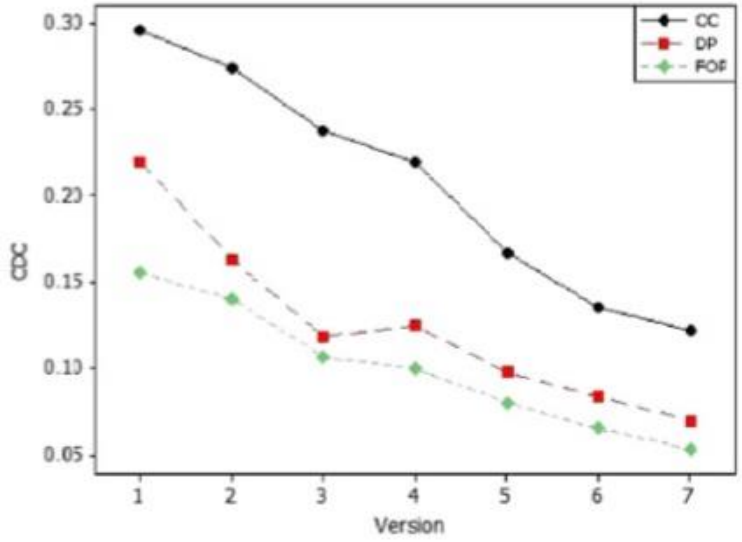


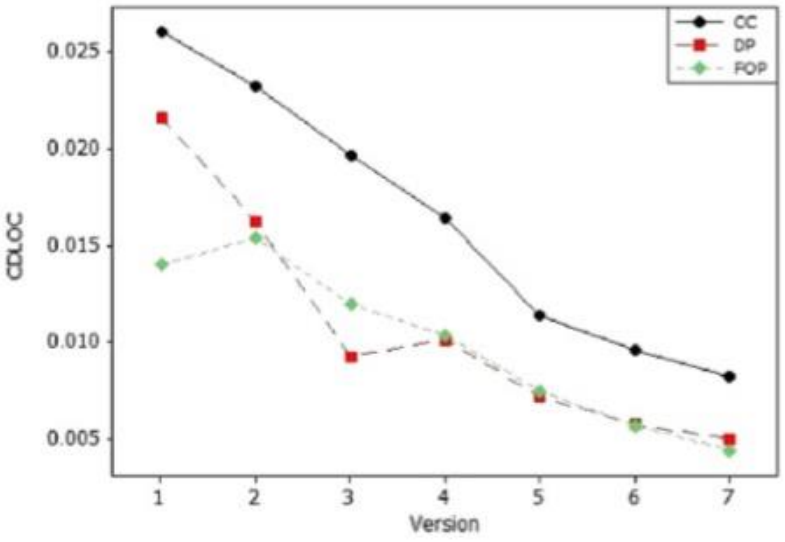


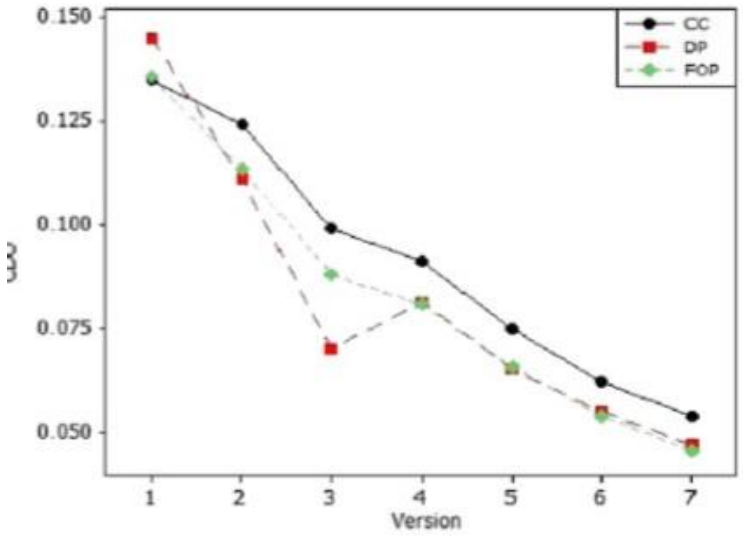


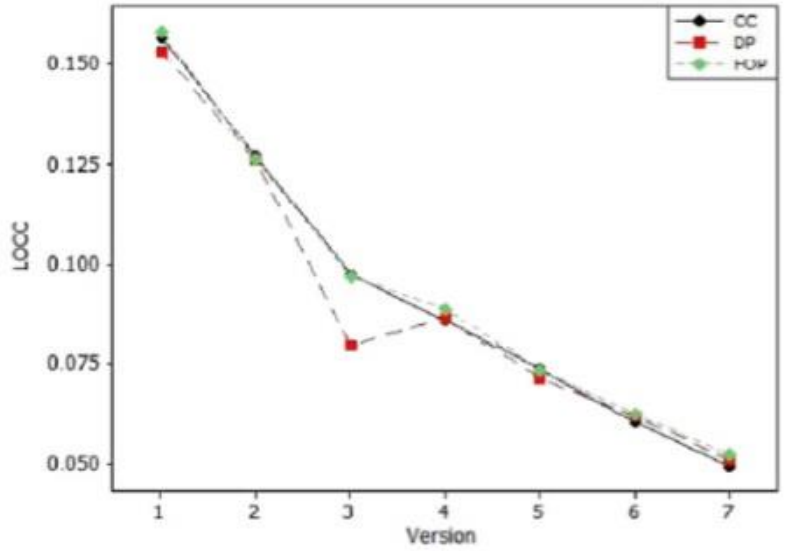


Metrics Values – MobileMedia









Cumulative Distribution Function

- Detailed analysis of the modularity metrics considering the dispersion of data
  - Minitab16 (empirical cumulative distribution function – ecdf)
    - Best fitted them data was 3-parameter Gamma
    - The data:
      - Not follow normal and symmetric distribution
      - Data values are typically concentrated in smaller values
  - Highest frequency of lower values

Discussion on modularity

- Interesting point
  - Despite of some differences
    - WebStore and MobileMedia presented an overall similar behavior in the four metrics.
    - CDO – FOP is better than CC and DP
    - CDLOC – FOP > CC > DP
    - CDO and LOCC – could not see significant differences between the three approaches in both systems. Nonetheless, it is possible to see a slightly better performance for FOP in both systems

Discussion on modularity

- Same global result can be observed in all versions
  - Higher is the version
    - Lower is the metrics values for all approaches
    - Lower is the difference between the approaches
- From the feature point of view
  - Some features to produce a similar behavior

RQ2

- Does the use of FOP provides more modular and stable design than CC and DP of the SPL features in evolution?
  - Yes, the data shows that FOP is better in this point

Discussion

- FOP succeeds in features with no shared code
  - Lower values and superior modularity
    - tangling (CDLOC) and scattering (CDC)
- When optional features are turned mandatory,
  - DP removal may cause the SPL architecture destabilization
  - DP < FOP
- Crosscuting features are problematic for all studied approaches

Discussion

- Ratio-based analysis of metrics tends to be less discriminative in larger systems
  - The larger is the evaluated software version
    - Lower are the metrics rations
    - Lower is the observable difference (approaches)
- On the use of a single variability mechanisms to construct SPLs
  - No silver bullet when it comes to mechanisms that manage variability in SPLs

Threats to Validity (1)

- Conclusion Validity

  - Measurement process

    - Independently checked
    - Spurious evidence – modularity metrics were indirectly used to answer RQ1

Threats to Validity (2)

- Internal Validity
  - Most analysis version of the SPLs were constructed by the authors
    - Different design options might have produced different results
  - Modularity Metrics
    - Depends on how accurate was the mapping of each concern to code elements

Threats to Validity (3)

- External Validity
  - SPLs may not represent all properties of real world systems
  - The evolutions scenarios may also not represent the large space of possibilities in real-world
  - Different results to other technologies/languages (Java and AHEAD)
  - Only Modularity and Change Propagation were considered

Construction Validity(4)

- How much support independent variables offer to produce robust answers
  - These metrics offer a limited view on the design stability and modularity problems
- Modularity metrics
  - Insufficient to allow a complete analysis of variability mechanisms with respect to SPL
- Change Propagation measures
  - Were used to complement the modularity analysis
    - They have learned that these two sets of metrics should not be analyzed in isolation

Concluding Remarks

- The development of an SPL has to provide means to anticipate changes
  - That is why incremental development has been largely adopted
  - FOP design tends to be more stable
    - When a change targets optional features
    - Class refinements adhere more closely the Open-Closed principle
  - DP and FOP strive to accommodate changes that require major restructuring

Future Works

- Study of different metrics and its relationship to other quality attributes in SPLs (robustness and reuse)
- Include Aspects
- A key challenge is to guarantee that only well-typed programs are generated
  - It is often hard, if not impractical
    - The approaches studied in this paper do not support modular type checking
      - Analyze the ability of each approach to deal with this problem

lab-soft
software engineering laboratory

Thanks!