On the use of feature-oriented programming for evolving software product lines — A comparative study

Gabriel Coutinho Sousa Ferreira, Felipe Nunes Gaia, Eduardo Figueiredo, Marcelo de Almeida Maia

Software Product Lines

- Mandatory features
- Variable features
  - Permit instantiation of different products by enabling or disabling specific SPL functionality
- Mechanisms to support variability management:
  - Feature Oriented Programming (FOP)
  - Conditional Compilation (CC)
  - Object-oriented design patterns
- Should minimize changes and should not degenerate modularity
- Open-Close principle: “Software should be open for extension, but close for modification”

Objective

- Evaluates comparatively three mechanisms for implementing variability on evolving SPL:
  - Conditional Compilation (CC)
  - Object-oriented design patterns (DP)
  - Feature-oriented programming (FOP)
- Five versions of WebStore and seven versions of MobileMedia

Conditional Compilation

- Preprocessor directives indicate pieces of code that should be compiled or not based on the value of preprocessors variables

```
1 private ControllerAction selectPaymentMethod(...) {
2     if (paymentType.equals("Default")) {
3       paymentAction = new GoToAction("payment.jsp");
4     }
5     //#if defined(Paypal)
6     if (paymentType.equals("Paypal")) {
7       paymentAction = new GoToAction("paypal.jsp");
8     }
9     //#endif
10      return paymentAction;
11 }
```

Object-oriented design patterns

- Dynamic binding and polymorphism mechanisms
- Decorator:
    - Provide entry point to add a feature behavior in a pluggable way

Feature-oriented programming

- Paradigm that consider features as a major abstraction
- Jak (AHEAD)
- Programs are constants and features are added to programs using refinements functions.

Objective

- Does the use of FOP has smoother change propagation impact than CC and DP during the evolving of an SPL?
- Does the use of FOP provides modular and stable design than CC and DP of the SPL features in evolution?

Methodology

1. Construction of two SPLs (MobileMedia and WebStore) using CC, DP and FOP
    a. WebStore was implemented from scratch.
    b. MobileMedia was already used in previous studies, so there is a full CC implementation. Only DP and FOP had to be implemented
2. Manual feature assignment of all produced code
3. Change propagation measurement and modularity metrics calculation
4. Quantitative and qualitative analysis of the result

WebStore

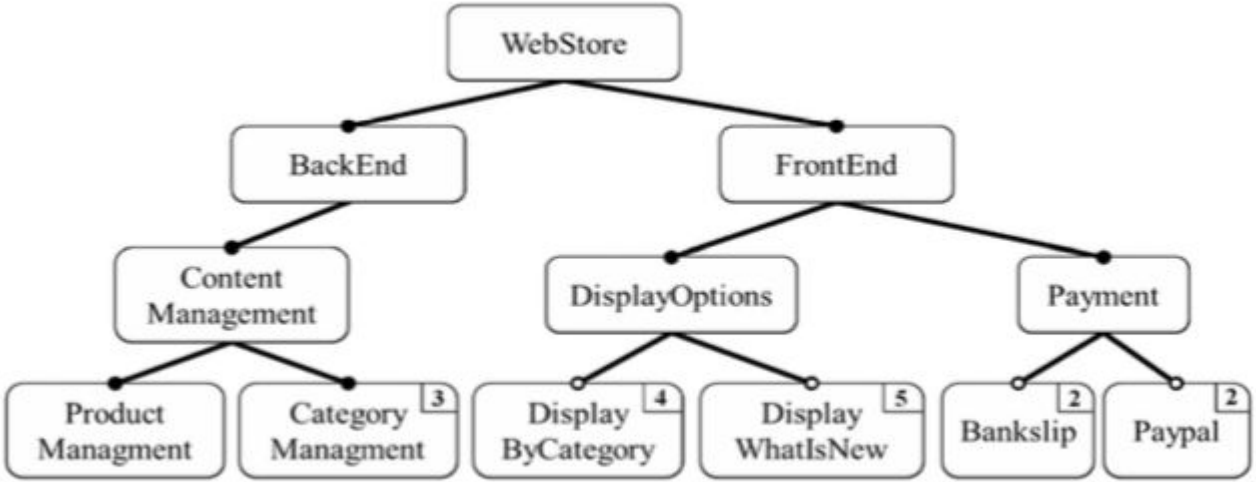

Fig. 1. WebStore basic feature model.

WebStore SPL implementation

	CC					FOP					DP				
	R.1	R.2	R.3	R.4	R.5	R.1	R.2	R.3	R.4	R.5	R.1	R.2	R.3	R.4	R.5
#Components	23	23	26	26	26	25	35	44	41	47	28	32	38	40	44
#Methods	138	139	165	164	167	150	170	200	198	208	142	147	175	177	182
LOC (approx.)	885	900	1045	1052	1066	945	1077	1257	1244	1303	915	950	1107	1121	1149

Summary of scenarios in WebStore.

Release	Description	Type of change	Extent of change
R1	WebStore core		
R2	Two types of payment included (Paypal and BankSlip)	Inclusion of optional feature	No extensive modification because the features can be well localized.
R3	New feature included to manage category	Inclusion of optional feature	Required changes in components related to Product and insertions of new components related to Category.
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	The inclusion of the new feature did not demand major modifications. Switching a feature from optional to mandatory required extensive removals in the DP.
R5	New feature included to display products by nearest day of inclusion	Inclusion of optional feature	Since this feature did not affect other functionalities, only minor changes and insertions were required.

MobileMedia

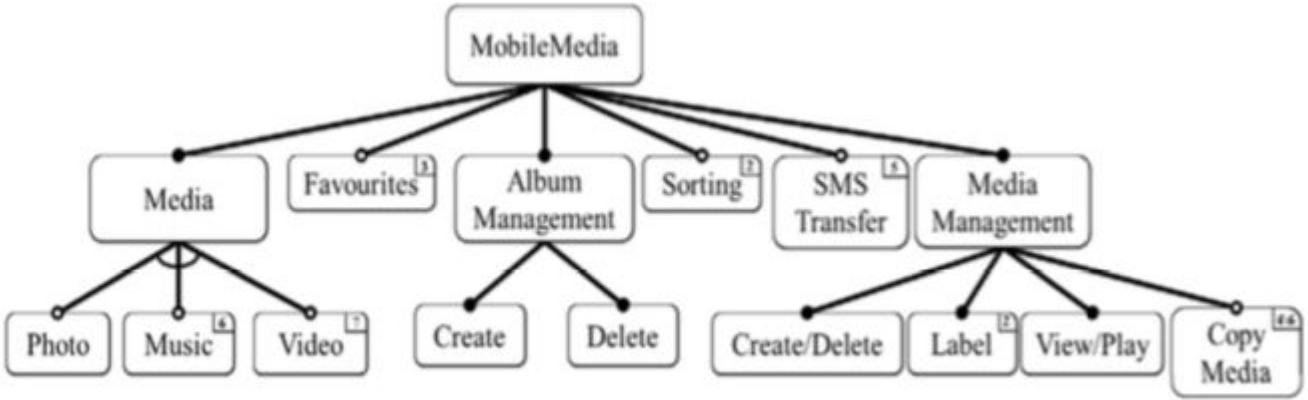

Fig. 2. MobileMedia basic feature model

MobileMedia SPL implementation.

	CC							FOP							DP						
	R.1	R.2	R.3	R.4	R.5	R.6	R.7	R.1	R.2	R.3	R.4	R.5	R.6	R.7	R.1	R.2	R.3	R.4	R.5	R.6	R.7
#Comp.	22	23	23	28	35	44	49	54	63	73	86	106	127	141	34	49	55	74	86	108	135
#Meth.	113	132	135	153	191	227	267	143	177	191	216	285	331	368	132	191	209	275	337	417	518
LOC	971	1147	1214	1380	1852	2334	2926	1142	1356	1458	1629	2163	2498	2827	1064	1430	1544	1936	2440	2952	3682

Summary of scenarios in MobileMedia.

Release	Description	Type of change	Extent 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	The feature Sorting required addition of new components and change components related to the use of this feature. For the feature EditLabel, a refactoring was conducted extracting a new PhotoController from the BaseController.
R3	New feature added to allow users to specify and view their favorite photos.	Inclusion of optional feature	The changes were narrowly localized.
R4	New feature added to allow users to keep multiple copies of photos.	Inclusion of optional feature	A major refactoring of BaseController was carried out producing four new specialized controllers.
R5	New feature added to send photo to other users by SMS.	Inclusion of optional feature	New controllers had to be included. New components related to SMS transfer had to be included. The SMSTransfer feature was designed as a specialization of the CopyPhoto 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	A major refactoring of PhotoController and PhotoListController was carried out producing two new generic media controllers. New controllers related to music operations had to be included.
R7	New feature added to manage videos	Inclusion of alternative feature	New controllers related to video operations had to be included.

Change propagation analysis

Summary of scenarios in MobileMedia.

			WebStore releases				Mobile media releases					
			R.2	R.3	R.4	R.5	R.2	R.3	R.4	R.5	R.6	R.7
Components	Added	CC	0	3	0	0	2	0	5	7	17	6
		FOP	4	6	2	4	10	10	23	21	74	14
		DP	10	9	8	6	15	6	20	13	79	29
	Removed	CC	0	0	0	0	1	0	0	0	8	1
		FOP	0	0	0	0	1	0	10	1	53	0
		DP	0	0	11	0	0	0	1	1	57	2
	Changed	CC	2	3	5	4	7	5	7	7	11	22
		FOP	1	1	0	0	10	6	23	10	28	13
		DP	4	4	4	1	13	11	29	11	11	27
Methods	Added	CC	1	26	0	3	22	3	37	38	103	47
		FOP	5	28	2	5	37	14	63	70	190	40
		DP	21	30	32	10	60	21	99	63	285	110
	Removed	CC	0	0	1	0	3	0	19	0	67	7
		FOP	0	0	0	0	3	0	38	1	144	3
		DP	1	0	34	0	1	3	33	1	205	9
	Changed	CC	2	2	6	2	9	7	10	7	26	30
		FOP	1	1	0	0	12	8	24	12	29	13
		DP	3	4	3	1	25	11	30	11	37	24
Lines of Code	Added	CC	15	148	7	14	197	67	538	478	1386	694
		FOP	35	160	14	28	243	102	490	551	1534	340
		DP	132	181	179	59	390	132	678	511	2189	820
	Removed	CC	0	3	0	0	21	0	372	6	904	102
		FOP	0	3	0	0	29	0	319	17	1199	11
		DP	0	1	192	0	24	18	286	7	1677	90
	Changed	CC	1	2	0	0	28	7	32	10	75	102
		FOP	1	2	0	0	21	10	83	8	62	19
		DP	9	2	3	0	45	13	85	12	75	46

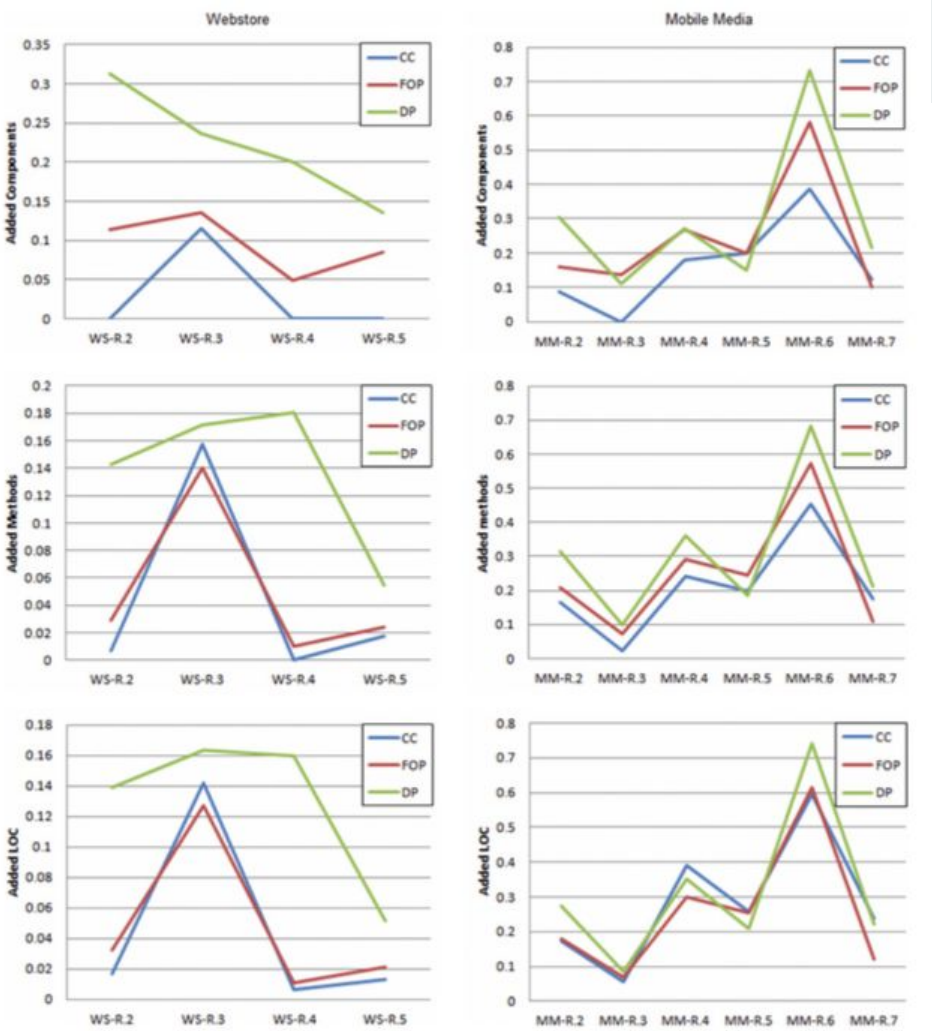

Fig. 3. Additions in WebStore and MobileMedia.

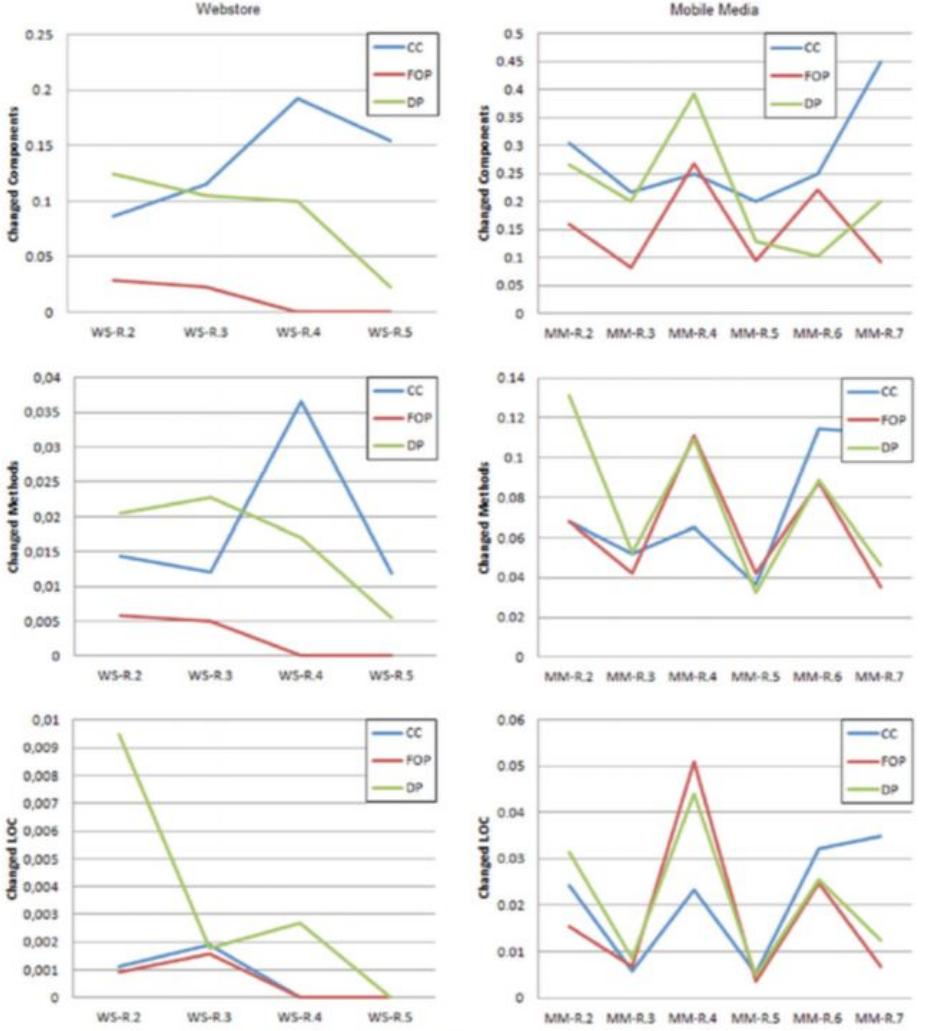

Fig. 4. Modifications in WebStore and MobileMedia.

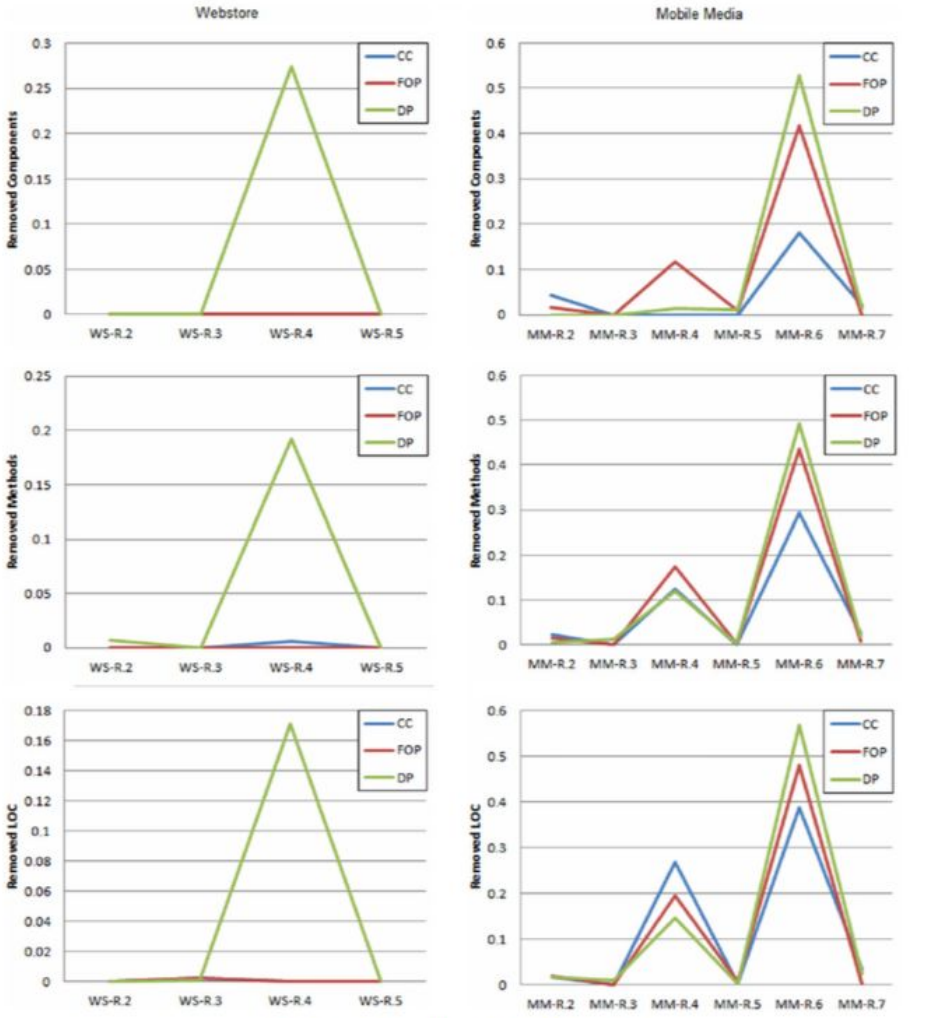

Fig. 5. Removals in WebStore and MobileMedia.

Change propagation analysis

- A lower number of modified and removed artifacts suggests a more stable solution
- Additions of artifacts indicates the conformance with the Open-Closed principle
- In general, CC mechanism presents lower number of added components and methods in both systems, compared to DP and FOP.
  - Insertions in CC have been carried out by modifying existent components instead of creating new ones
- For MobileData there is no sharp difference between the measures of the three mechanisms.
  - Product configuration in DP-based solutions are done at runtime
- WebStore has higher number of components, methods and lines of code with DP, than with FOP and CC.

Discussion

- CC releases have consistently lower number of added components than DP and FOP.
- FOP and DP strive to accommodate changes that require major features restructuring and usually demand a greater amount of component removal.
- Based on component insertion, CC does not adhere to Open-Closed principles as FOP and DP adhere.
- FOP and DP mechanisms have not a significant difference, because in WebStore, DP introduces more components and in MobileMedia the inverse situation occurs.

Modularity analysis

- Metrics for quantifying feature modularity:
  - CDC - Concern Diffusion Components
  - CDO - Concern Diffusion over Operations
  - CDLOC - Concern Diffusion over Lines of Code
  - LOCC - Lines of Concern Code

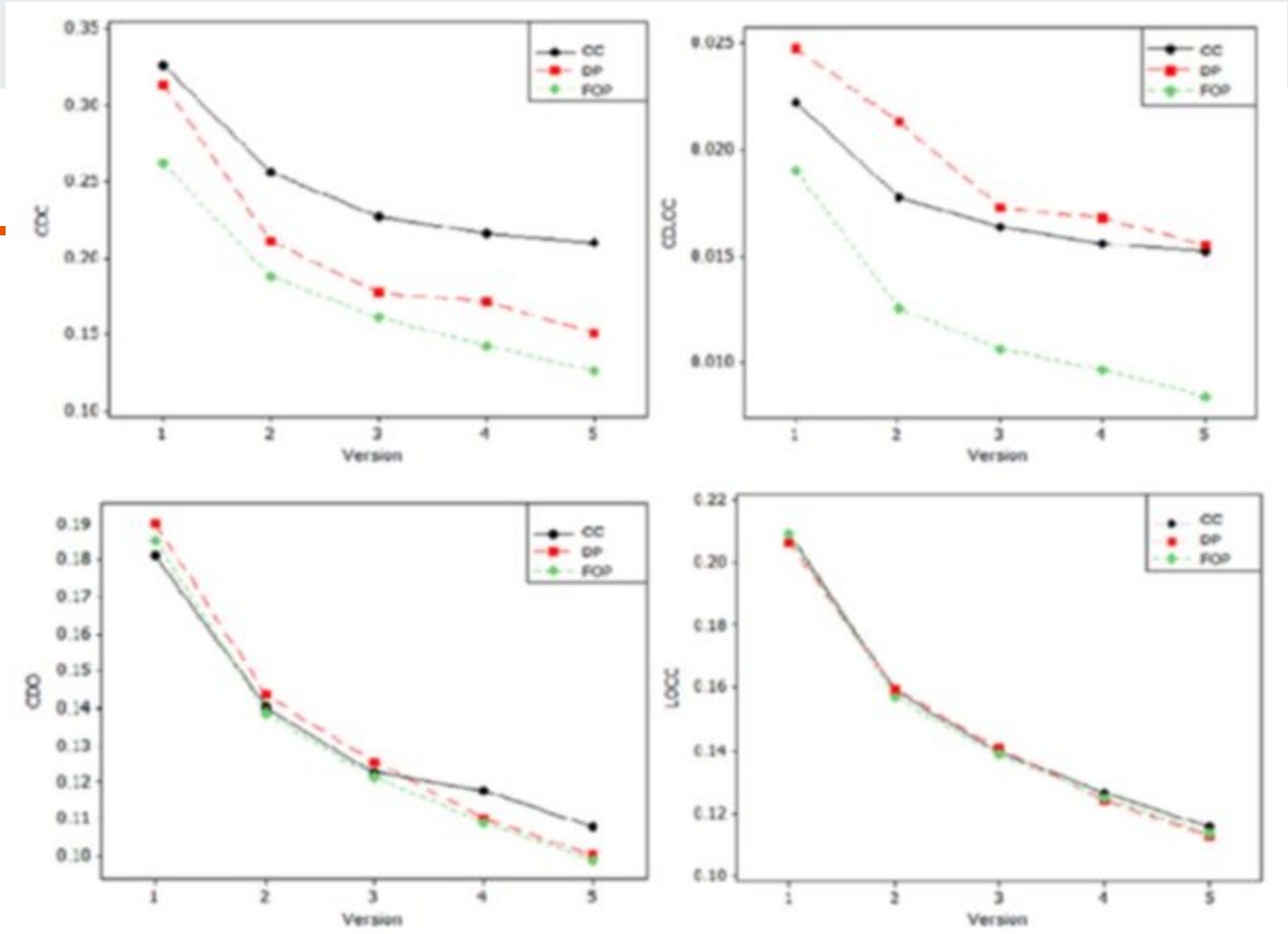

Fig. 6. Metrics values through WebStore evolution.

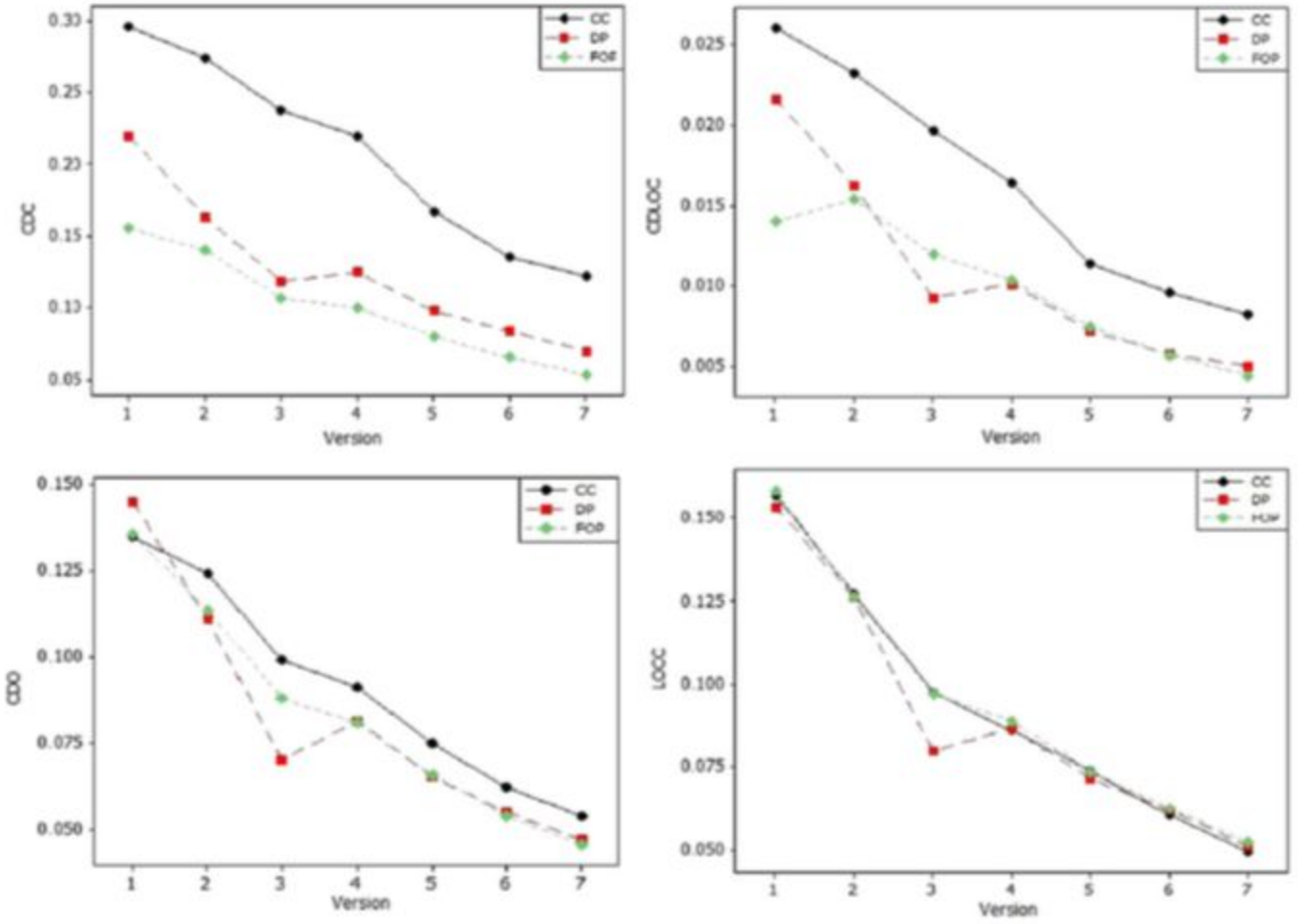

Fig. 7. Metric values through MobileMedia evolution.

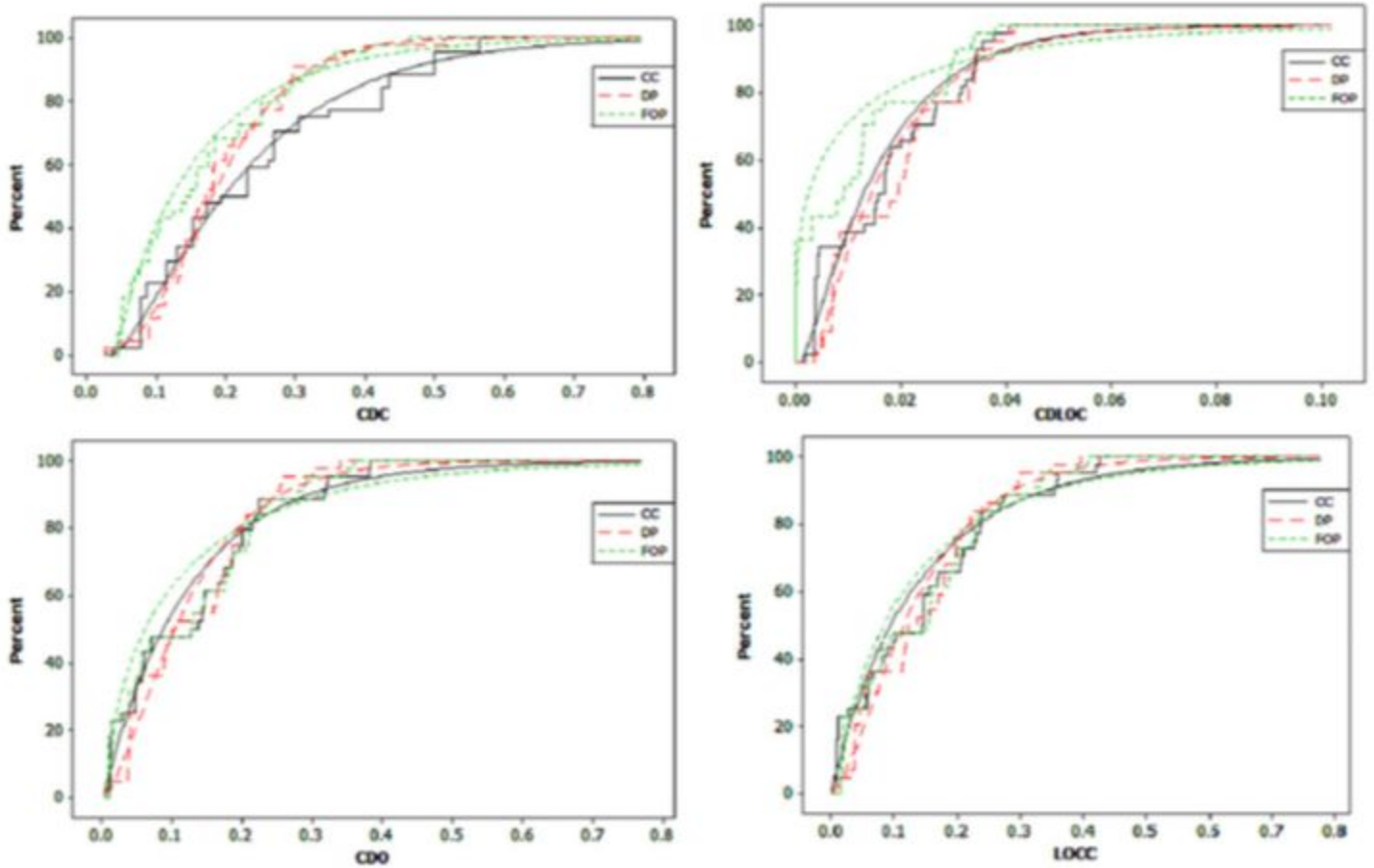

Fig. 8. Empirical CDF for all versions of Webstore (3-parameter Gamma).

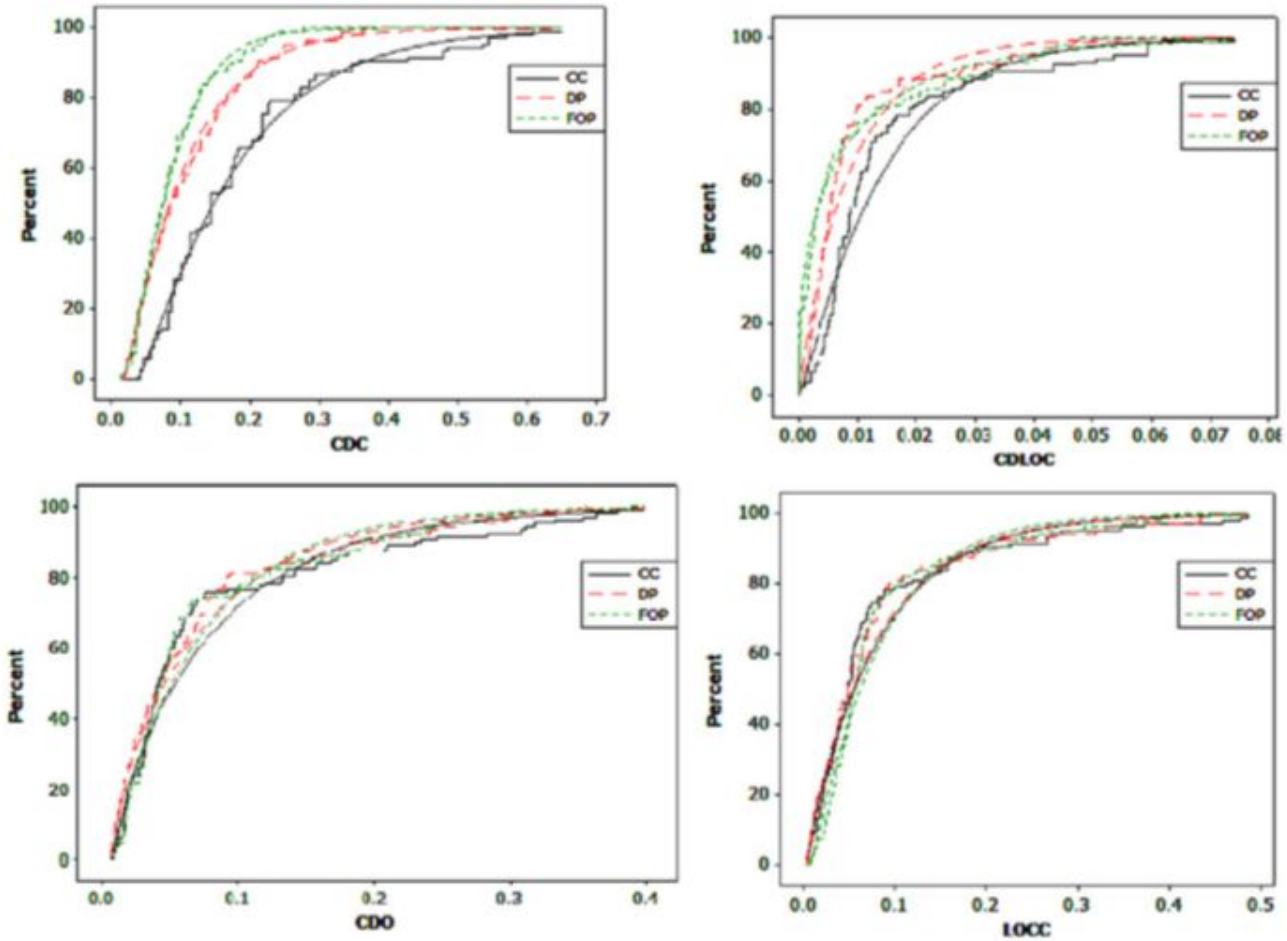

Fig. 9. Empirical CDF for all versions of Mobile Media (3-parameter Gamma).

Conclusion

- WebStore and MobileMedia are special purpose systems that may not represent all properties of real world systems.
- The evolution scenarios may also represent the large space of possibilities in real-world SPL evolution scenarios. The paper not investigate some intricate situations involving feature interaction that may appear in large SPLs.
- Only the Java programming language and the AHEAD environment were considered on the study. The result can be different if other languages and environments were used.
- Only modularity and change propagation metrics were considered helpful to point out the vulnerability mechanisms benefits. However, they provide only a limited view of these benefits, as they do not measure the real effort required to perform SPL changes.

Conclusion

- The work main contributions were the development of an open benchmark for the evaluation of evolving SPLs, a qualitative and quantitative data analysis framework and an extensive data analysis of collected metrics using the benchmark and the framework.
- FOP tends to be more stable than the other traditional widely-used approaches
- FOP class refinements adhere more closely the Open-Closed principle.
- The results indicate that conditional compilation (CC) have not be adequate when used in evolving SPLs when feature modularity is a major concern.
- DP and FOP also strive to accommodate changes that require major restructuring.
- The work revealed evidences that FOP still has drawbacks that require the combination with others mechanisms.