

Incremental approach to Software Product Line Engineering

Ștefan Stănciulescu
IT University of Copenhagen

VARIETE Workshop
18th November 2013

Joint work with University of Waterloo, Canada,
University of Koblenz-Landau and TU Braunschweig, Germany

Agenda

1. Software evolution
2. Evolution in software product lines
3. Challenges and goals
4. Idea and work in progress
5. Conclusion and further work

Software evolution

Software evolution

Evolution

- New features

- Requirements change

- Bug fixes

Subsequent products are developed by cloning previous ones

Different reasons : costs, time, organizational issues, etc.

Maintaining similar products becomes expensive

Software evolution

Adopt a Software Product Line (SPL)

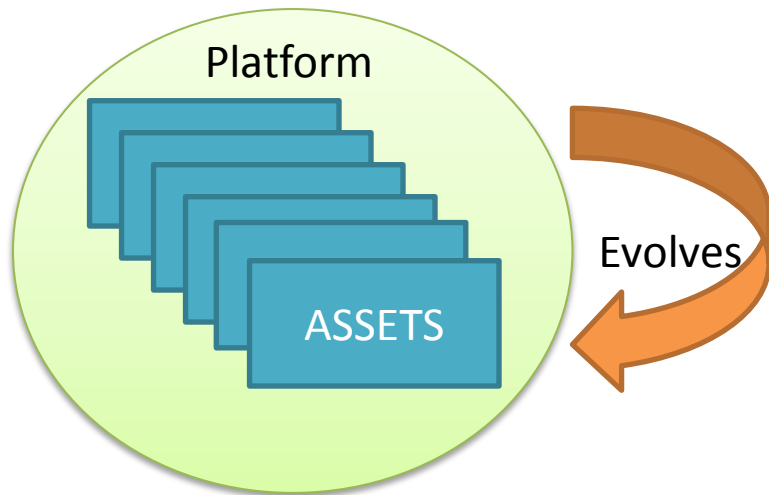
Time and resource consuming task

Organizational issues

Transition phase

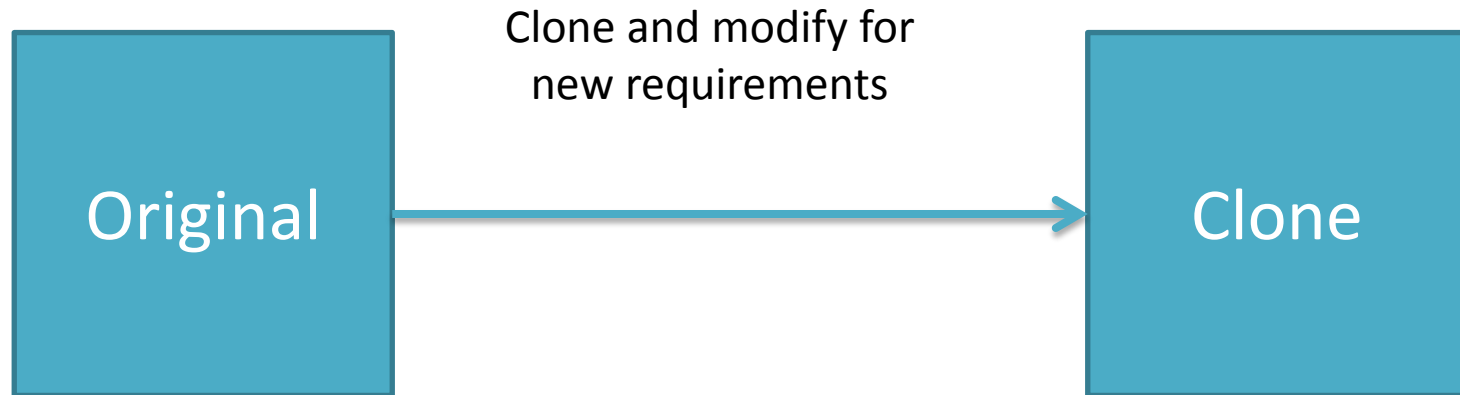
Evolution in software product lines

Evolution in product lines



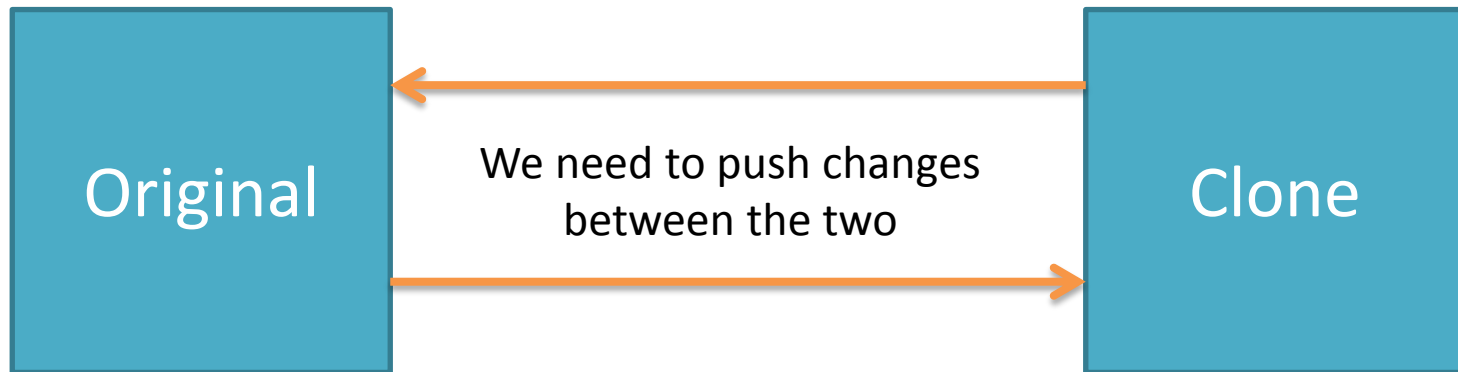
Evolution in product lines

Clone a product



- ✓ Fast
- ✓ Less costs and efforts
- ✓ Easy

Evolution in product lines



SPL and Cloning

SPL

Sharing is explicit
Easy to derive new products
Control over the process

Difficult to introduce a SPL
Organizational challenges
Transition phase
Difficult to prototype

Clone

Easy
Fast
Trusted
Widely used

No control
Hard to propagate changes
No tracking between clones

Challenges & Goals

Challenges

Clones introduce double management problem

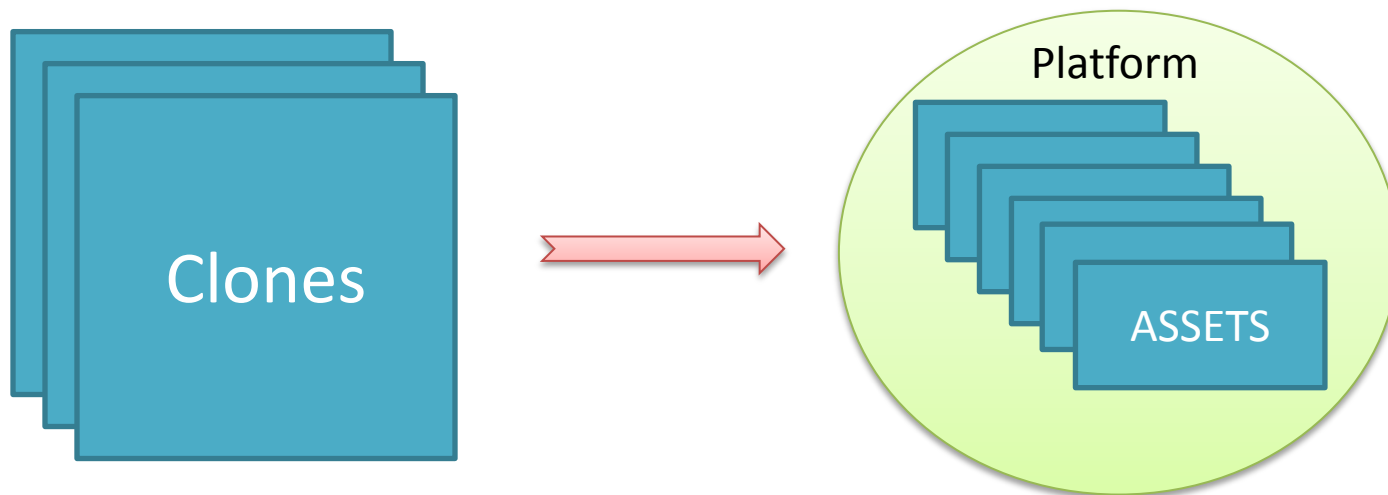
Hard to propagate changes

Gain SPL benefits during development

Migrate to SPL without transition phase

Goals

1. Manage the cloning mechanism
2. Develop tools for this
3. Be able to easily create a platform



Idea and work in progress

Idea - *Virtual* platform via metadata

The core is based on **metadata**

Operators use metadata

A combination of operators implements a tool for a **scenario**:

- Create clone
- Add/Remove features
- Propagate changes/features
- Many others...

Idea - *Virtual* platform via metadata

Applying a scenario tool produces complete **metadata**

The collection of metadata forms the **virtual platform**

Later use the **metadata** to build the real platform

Incremental approach towards a SPL

1. Develop products using the clone and own approach

Introduce *virtual platform*:

2. Start using tools and record metadata

- Create a feature model
- Map features to artifacts

3. Use the *virtual* platform to create new products

4. Build the SPL platform

Incremental approach towards a SPL

1. Develop products using the clone and own approach

Introduce **virtual platform**:

2. Start using tools and record metadata

- Create a feature model

- Map features to artifacts

3. Use the *virtual* platform to create new products

4. Build the SPL platform

Incremental approach towards a SPL

1. Develop products using the clone and own approach

Introduce **virtual platform**:

2. Start using tools and record metadata

- Create a feature model
- Map features to artifacts

3. Use the ***virtual*** platform to create new products

4. Build the SPL platform

Incremental approach towards a SPL

1. Develop products using the clone and own approach

Introduce **virtual platform**:

2. Start using tools and record metadata

- Create a feature model
- Map features to artifacts

3. Use the *virtual* platform to create new products

4. Build the SPL platform

Incremental approach towards a SPL

1. Develop products using the clone and own approach

Introduce **virtual platform**:

2. Start using tools and record metadata

- Create a feature model
- Map features to artifacts

3. Use the *virtual* platform to create new products

4. Build the SPL platform

Benefits

Incremental return

No transition phase

Gain SPL benefits with minimal efforts

Clone-based, so it is easy to adopt

Work in progress

Feature related scenarios

Add/Remove feature

Copy feature

Modify feature

Merge features

Other scenarios

Work in progress

Project related scenarios

Clone product

Merge products

Notify of changes

Work in progress

Operators:

- *findFeature* – finds the feature's name, description, artifacts
- *findDEP* – finds the dependencies of the feature
- *findCST* – finds constraints of the feature
- *linkProducts* – links two products (traceability)
- *linkFeature* – maps a feature to artifacts
- *findChanges* – finds all changes related to this feature

Related work

A framework for managing cloned product variants [1]

Migration to SPL – Danfoss study case [2]

[1] J. Rubin and M. Chechik. A framework for managing cloned product variants. In Proceedings of the 2013 International Conference on Software Engineering, ICSE '13

[2] H. P. Jepsen, J. G. Dall, and D. Beuche. Minimally invasive migration to software product lines. In Proceedings of the 11th International Software Product Line Conference, SPLC '07

Conclusion & Further work

Cloning is fast, easy and used

Provide tool support for systematic cloning

Virtual platform using metadata

Future work:

Define and implement operators and metadata