



Cloning Practices: Why Developers Clone and What can be Changed

Author : Gang Zhang, Xin Peng, Zhenchang Xing, Wenyun Zhao

2012 28th IEEE International Conference on Software Maintenance (ICSM) 285-294



目次

- 背景, 目的
- アプローチ
- アンケート, インタビューの回答
- まとめ, 所感



背景, 目的

背景

- 開発者がクローンの生成や削除をする理由について分かればクローンに対してどのような操作をすべきか分かるようになる
- クローンを生成する理由を調査した研究はいくつかあるが、クローンの様々な操作に関する理由の詳しい調査が行われていない。

目的

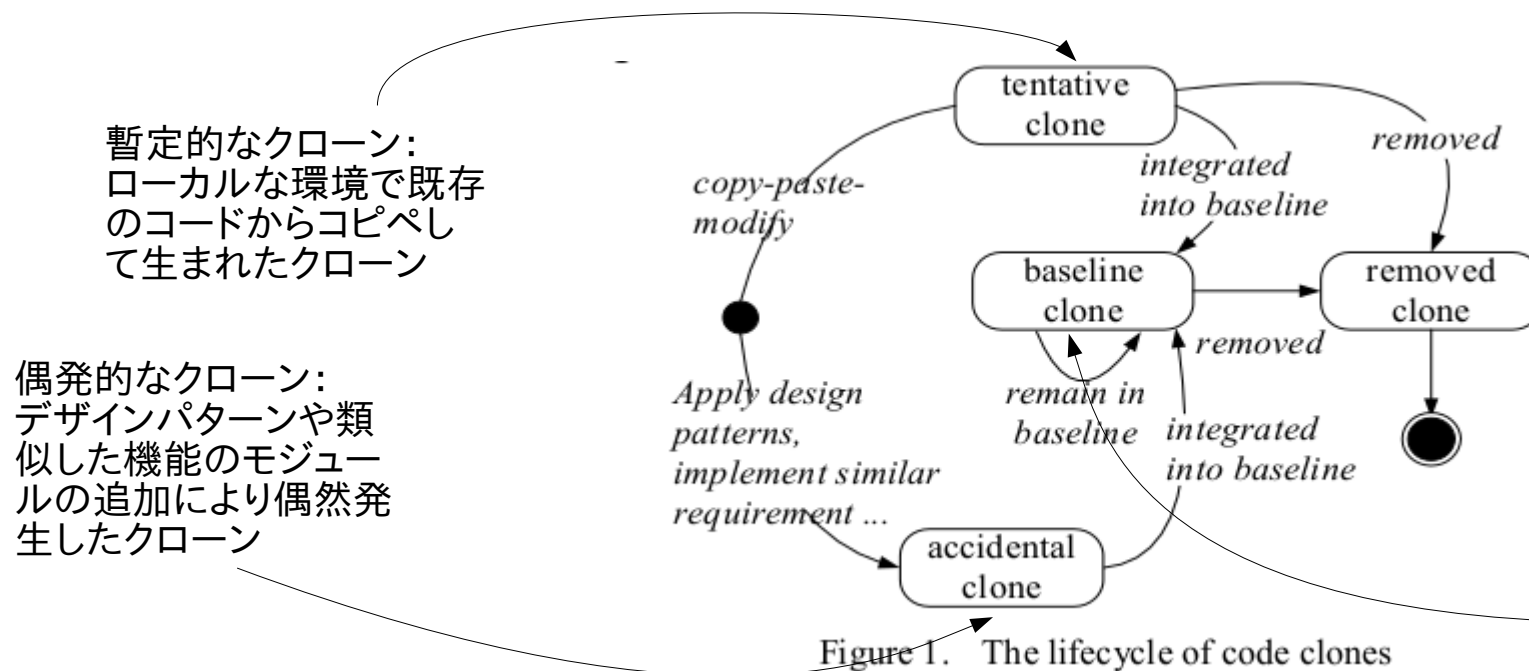
- 技術的, 組織的, 個人的な観点からクローンの生成や削除などを行う理由を調査し, より詳細な理由を明らかにする



アプローチ

クローンのライフサイクル

- クローンが生成されてから削除されるまでの流れを表したものの



ベースラインクローン:
暫定的なクローンや偶
発的なクローンがバー
ジョン管理システムの
主要な部分で統合さ
れたクローン

クローンの収集

- 本研究の筆者の 1 人が開発に携わる C/C++ で書かれた商用システムからクローンを収集
- CCFinderX[5] を用いて 47,527 個のクローン集合 (286,983 個のクローン片) を抽出
- [24] での研究の手法を用いて 80 個のモジュール単位のクローン集合 (417 個のモジュール) を抽出
- 以上のクローンの中から異なるサイズのクローン集合 141 個を手動でサンプリング

[5] T. Kamiya, S. Kusumoto, and K. Inoue. CCFinder: A multilingual token-based code clone detection system for large scale source code. IEEE Trans. on Soft. Eng., Vol. 28(7): 654- 670, 2002.

[24] H. A. Basit, S. Jarzabek: A data mining approach for detecting higher-level clones in software. IEEE Trans. Soft. Eng., Vol. 35(4): 497-514, 2009.

アンケート

- サンプルングしたクローンに関わる開発歴の異なる開発者(3 ~ 12 年)の中から 21 人選択
- アンケートの回答からカテゴリーを作り, それぞれの回答をそれぞれのカテゴリーに分ける

アンケートの内容

TABLE 1. QUESTIONNAIRE DESIGN

start → tentative clone	q1. Based on my experience, I clone most often during: _____
	q2. I <u>seldom</u> <u>sometimes</u> <u>often</u> copy-paste-modify code
	q3. I copy-paste-modify code because: _____
start → accidental clone	q4. I <u>seldom</u> <u>sometimes</u> <u>often</u> notice code clones accidentally
	q5. Based on my experience, accidental code clones are often introduced because _____
tentative clone → removed clone	q6. I <u>seldom</u> <u>sometimes</u> <u>often</u> remove tentative clones that I created before they enter the baseline of the system
	q7. If you have ever removed some tentative clones, what are your considerations for the removal? _____
tentative clone → baseline clone	q8. I <u>seldom</u> <u>sometimes</u> <u>often</u> leave tentative clones unattended, and thus they become baseline clones in the baseline of the system
	q9. Why do not you remove tentative clones? _____
baseline clone	q10. I <u>seldom</u> <u>sometimes</u> <u>often</u> encounter code clones in baseline versions of the system
	q11. I <u>seldom</u> <u>sometimes</u> <u>often</u> feel uncomfortable with code clones in baseline versions of the system
baseline clone → removed clone	q12. I am <u>seldom</u> <u>sometimes</u> <u>often</u> willing to remove code clones in baseline versions of the system
	q13. I am aware of the following techniques for removing code clones _____
	q14. I <u>seldom</u> <u>sometimes</u> <u>often</u> remove baseline code clones during development
	q15. If you have ever removed some code clones from baseline versions of the system, what are your considerations for the removal? _____
baseline clone → baseline clone	q16. I <u>seldom</u> <u>sometimes</u> <u>often</u> remove code clones in baseline versions that I think are harmful
	q17. What makes you not remove harmful baseline clones? _____

クローンのライフサイクルのフェーズごとに質問を作成

インタビュー

- アンケートの回答から理由を更に深掘りするためにインタビューを行う
- 5why 形式のインタビューを行う
 - 5why : 聞き出した回答に対して質問をするのを 5 回繰り返す手法



アンケート, インタビューの回答

回答の整理

- アンケートの形式を基に得られた回答を以下の質問に対する回答としてまとめる
 - どれほどの頻度でクローンを生成し, クローンに対する姿勢はどのようなものか
 - なぜ暫定的なクローンを生成したか
 - なぜ暫定的なクローンをベースラインに統合したか(しなかったか)
 - なぜベースラインクローンは取り除かれたか(取り除かれなかったか)
 - なぜ偶発的なクローンを生成したか

どれほどの頻度でクローンを生成し, クローンに対する姿勢はどのようなものか

- アンケートの質問 2 , 10* の回答に関するもの
- 頻度については 33% の割合で頻繁に, 67% の割合でたまに行うということになった
- 姿勢については以下の表のようになった

TABLE 2. DEVELOPERS' ATTITUDE/BEHAVIOR ON CLONE

cluster id	count (ratio)	characteristics		
		attitude	skill	behavior
A	3(17.6%)	strong	very good	often
B	3(17.6%)	strong	good	sometimes
C	11(64.8%)	middle	middle	sometimes

*q2. I seldom|sometimes|often copy-paste-modify code ____

q10. I seldom|sometimes|often encounter code clones in baseline versions of the system

なぜ暫定的なクローンを生成したか

- Following existing solutions と Lack of proper framework の回答については、フレームワークの不備によってクローンを生成せざるを得ない状況があったため比較的大きな割合となっている

TABLE 3. REASON FOR INTRODUCING TENTATIVE CLONES

reason	count (percentage)
Following existing solutions: There already exists a solution for the problems to be solved.	10 (47.6%)
Avoiding breaking existing features: Reusing needs to change existing code which may break existing features.	9 (42.9%)
Difficult to be reused: The original code contains content irrelevant to the task at hand.	9 (42.9%)
Duplicate code is short: Clone is not harmful if duplicated code is short.	7 (33.3%)
Lack of proper framework: Has to copy-paste-modify due to lack of proper framework.	7 (33.3%)
Easy for separate evolution: Duplicate code so that they can be evolved independently.	1 (4.8%)
Time limitation: Pressured by project progress or so.	10 (47.6%)
Issues of code ownership: Code to be reused is owned by other developers, teams, or even organizations.	5 (23.8%)
Learning and discovery: Copy-paste-modify for experiment or learning purpose.	9 (42.9%)

白い部分: 技術的な理由
グレーの部分: 組織的な理由
濃いグレーの部分: 個人的な理由

q3. I copy-paste-modify code because: ____

なぜ暫定的なクローンをベースラインに統合したか (しなかったか)

- 統合しなかった理由としてはコードの質を下げないためであるという回答
- 統合された組織的な理由として時間の制限やリスクの回避が多く挙げられている

TABLE 5. REASON FOR TENTATIVE CLONE BECOMING BASELINE CLONE

reason	count (percentage)
Difficult to change existing code: tentative clones becomes difficult to remove when developer want to do so.	8 (38.1%)
Not considered harmful: For some clones, developers do not think they are harmful.	6 (28.6%)
Be consistent with existing clones: Because there are already clones for the similar problem, developer has to keep consistent with them.	5 (23.8%)
Time limitation: tentative clones enter baseline because there is no time to remove them.	14 (66.7%)
Risk avoidance: in the later phase of the project, developers are afraid that changing source code may break the system.	14 (66.7%)
Not my responsibility: Developer does not think they have responsibility to prevent baseline clones.	4 (19.0%)
Performance evaluation by LOC: code lines produced is used as a productivity measurement.	1 (4.8%)
No awareness of harmfulness of clone: developers lack of awareness of how and why clones can be harmful.	7 (33.3%)
Knowledge/skill limitation: Developers do not know how to remove clones.	6 (28.6%)

q7. If you have ever removed some tentative clones, what are your considerations for the removal? ____

q9. Why do not you remove tentative clones? ____

なぜベースラインクローンは取り除かれたか (取り除かれなかったか)

- 取り除かれた理由はコードの質を下げないため
- 取り除かれなかった理由の技術的な理由として割合が大きい理解の二度手間を避けるためというものがある
- よく知れ渡っているクローンに関しては、取り除いてしまうとコードの理解度に影響を与えてしまうと考えられる

TABLE 7. REASON FOR NOT REMOVING BASELINE CLONES

reason	count (percentage)
Avoiding re-learning: Developer are already familiar with cloned codes.	16 (76.2%)
No test automation: no test automation to protect the code that would be affected by clone removal.	14 (66.7%)
Keeping solution consistent: Due to the existence of many copies, it is impossible to remove all of them, while removing some of them will introduce inconsistencies.	10 (47.6%)
Avoiding breaking existing features: Clone removal needs change to existing code that may break existing features.	10 (47.6%)
Difficult to change existing code: the cloned codes become difficult to remove when developer want to do so.	8 (38.1%)
Lack of tool support: no proper tool support for clone analysis or removal.	8 (38.1%)
Lack of initial and historical context: there is little or no information about why clones are introduced and how they evolve to current state.	4 (19.0%)
Risk avoidance: removing duplication will increase the project risk.	17 (81.0%)
Cost limitation: no budget in the project for clone removal.	11 (52.4%)
Time limitation: no time for clone removal.	11 (52.4%)
Avoiding cascading effects in the project: clone removal may create additional tasks (e.g. communicating with other developers).	10 (47.6%)
Performance evaluation: not related to my performance evaluation	7 (33.3%)
Not appreciated: clone removal is not appreciated by organization even the developers take initiative.	7 (33.3%)
No plan: clone removal is not listed as a target of the project.	7 (33.3%)
Not my responsibility: the developers do not feel that it's their responsibility to remove others' clone.	5 (23.8%)
Knowledge/skill limitation: Keeping the clone is the only choice which developer can use because developers do not know how to remove clones.	4 (19.0%)
Not interested: developers are more interested in writing new code than removing clones.	2 (9.5%)

q10. I seldom|sometimes|often encounter code clones in baseline versions of the system

q17. What makes you not remove harmful baseline clones? ____

なぜ偶発的なクローンを生成したか

- 既存の解決法の認識不足, 自動生成されたコード, 不完全なクローンのマージの3つが主な理由として挙げられる
- 既存の解決法の認識不足は技術的な理由と採れるが, 組織全体で既存の解決法を伝えきれていなかったことから, 組織的な理由とも採れる



まとめ, 所感

まとめ

- 開発歴のある開発者しかクローンを取り除こうとしない
- フレームワークなどの不備により, クローンを作らざるを得ない場合がある
- ベースラインにコードクローンが入り込んだり, 取り除かれたりしない理由は時間の制限やリスクの回避のためである

所感

- クローンに対する現場の声を聞くことで、開発者がクローンについてどう思っているかを少しは知ることが出来た
- クローンを取り除くことが出来るのは熟練者だけということから、あまり開発歴の無い開発者はクローンを取り除くのは怖いことであることが想像できた