Time-Related Patterns of Schema Evolution

EDBT 2025

Panos Vassiliadis, Alexandros Karakasidis

http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

University of Ioannina, Hellas

University of Macedonia, Hellas

In Memoriam

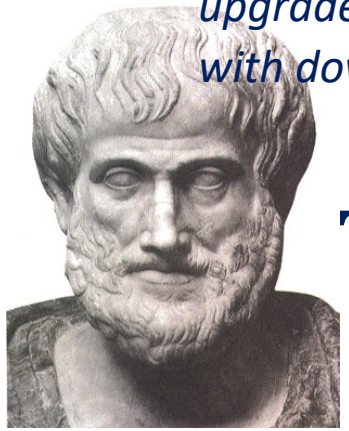
Matthias Jarke
(1951 - 2024)

Thank you for everything!

The dreaded schema evolution

- Stonebraker at al., CACM 60, 1 (2017): "*In a survey of 20 database administrators (DBAs) at three large companies in the Boston area, we found that . . . , **DBAs try very hard not to change the schema when business conditions change, preferring to "make things work" without schema changes**. If they must change the schema, they work directly from the relational tables in place. "*

- Limoncelli CACM 62, 1 (2019): "***When the software is tightly coupled to the database schema it becomes impossible to perform software upgrades that require a database schema change**. If you first change the schema, the instances will all die or at least get confused by the change; . . . Why not upgrade the instances first? Sadly, as you upgrade the instances' software one by one, the newly upgraded instances fail to start as they detect the wrong schema. You will end up with downtime until the schema is changed to match the software*"



The nature that needs change is vicious; for it is not simple nor good…

Nicomachean Ethics, Book VII, Aristotle

What are the laws of database schema evolution?

Wouldn't it be nice to have (Long term research goals)

- ❑ … a set of "**laws**" (patterns of repeated behavior under specific contexts) on how database schemata change?
- ❑ … a **theory / model** to explain them?
- ❑ … a set of **schema design patterns and anti-patterns** to make evolution easier?
- ❑ … similarly: **software design patterns and anti-patterns**
- ❑ … **prediction mechanisms** as part of prj management?
- ❑ … a set of **education guidelines** on to how teach this?

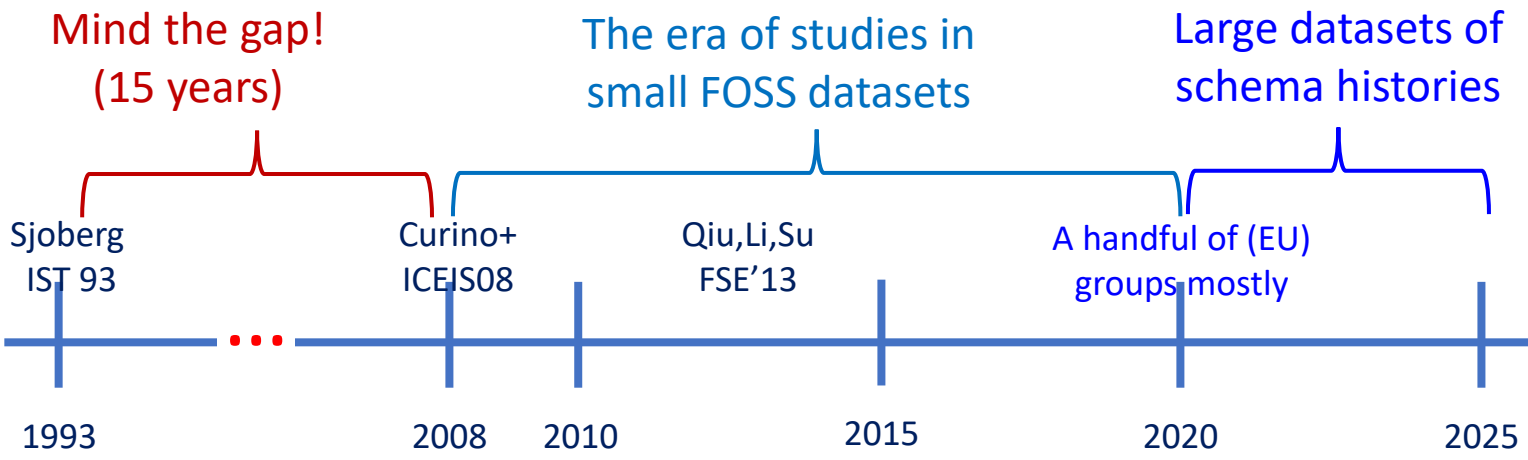
In this talk, I will very quickly give summaries of …

- Why we are not close to a solution of the problem
- How I have attacked the problem in the last years
- The **availability of data, tools and methods** to study schema histories
- **Recent findings presented in the EDBT ’25 paper**
- **Take-away thoughts**

Why aren't we there yet?

- Problem #1: **we, as a community, don't care enough!**
  - we build things and don't look back on what happens with them
- Problem #2: **we ~~don't~~ did not have the data!**
  - Historically, nobody from the research community had access + the right to publish to version histories of database schemata
  - Open-source tools internally hosting databases have changed this landscape & allowed first to work on small collections of schema histories && later, on larger ones

- Problem #3: **we have not "agreed" on concepts / problems / methods**

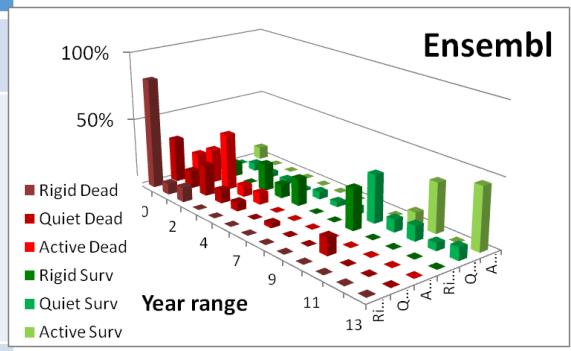


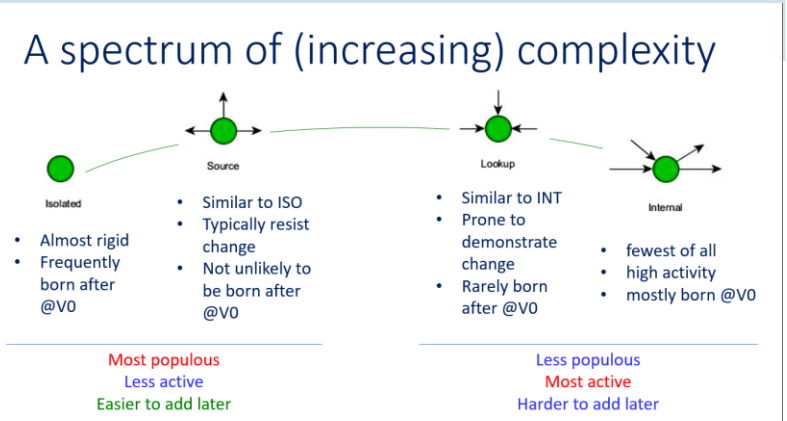
Shout-out to colleagues at Univ. Namur, Murcia, && Regensburg

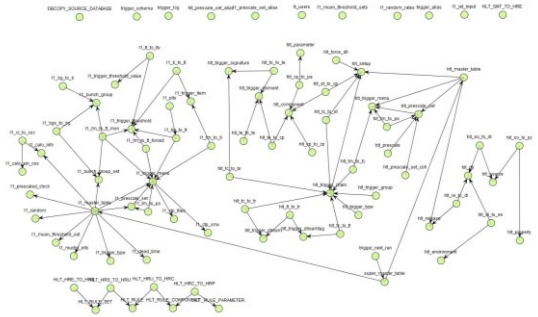
See the paper for **related work**

In our work in the 10's, the lack of large schema histories has been a major pain…

Univ. Ioannina	Results on few selected schema histories
ER15, IS17	First patterns on **how tables live, die & change**
CAiSE 16 + JoDS 17	The **Electrolysis** pattern: Survivors, mostly long-lived (esp. active ones) & quietly active are radically different than dead tables, being mostly short-lived & rigid!
ER'17, ER'20	How (the frequently absent) **Foreign Keys evolve**: FKs don't change too much, and often die. Spectrum of change propensity wrt in- & out-degrees

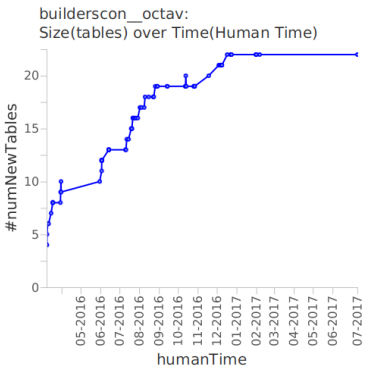




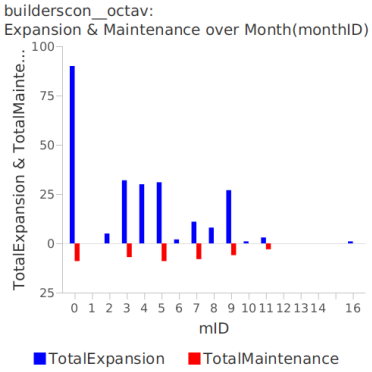


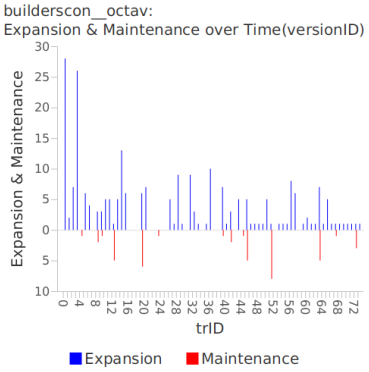
… so I did sth about it [ICDE 21]

- [ICDE21] Compiled a **large data set of 195 representative schema histories from FOSS projects**

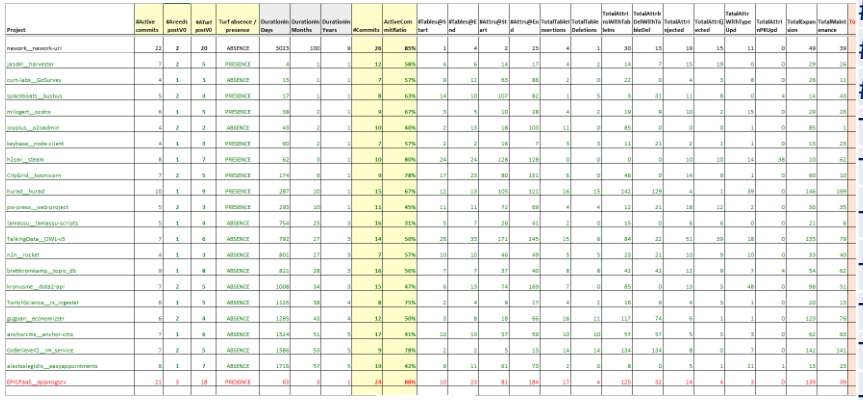


- [EDBT 23] Studied src & schema co-evolution. Schema evo is premature: most times schema evo precedes both src & time





- [EDBT 25] We isolated **151 prjs** with duration larger than a year
- We **extracted change patterns in time both quantitatively and qualitatively**



Project
#Active commits
#Areeds postV0
#ATurf postV0
Turf
Ratio
Turf absence / presence
DurationInDays
DurationInMonths
DurationInYears
#Commits
#Tables@Start
#Tables@End
#Attrs@Start
#Attrs@End
TotalTableInsertions
TotalTableDeletions
TotalAttrInsWithTableIns
TotalAttrbDelWithTableDel
TotalAttrInjected
TotalAttrEjected
TatalAttrWithTypeUpd
TotalAttrInPKUpd
TotalExpansion

https://github.com/DAINTINESS-Group/Schema_Evolution_Datasets/

We now have both tools and data on schema evolution

Everything is online!

My group's git page

https://github.com/DAINTINESS-Group/

has links to **Data sets**

https://github.com/DAINTINESS-Group/Schema_Evolution_Datasets/tree/master/SchemaEvolutionDatasets2020

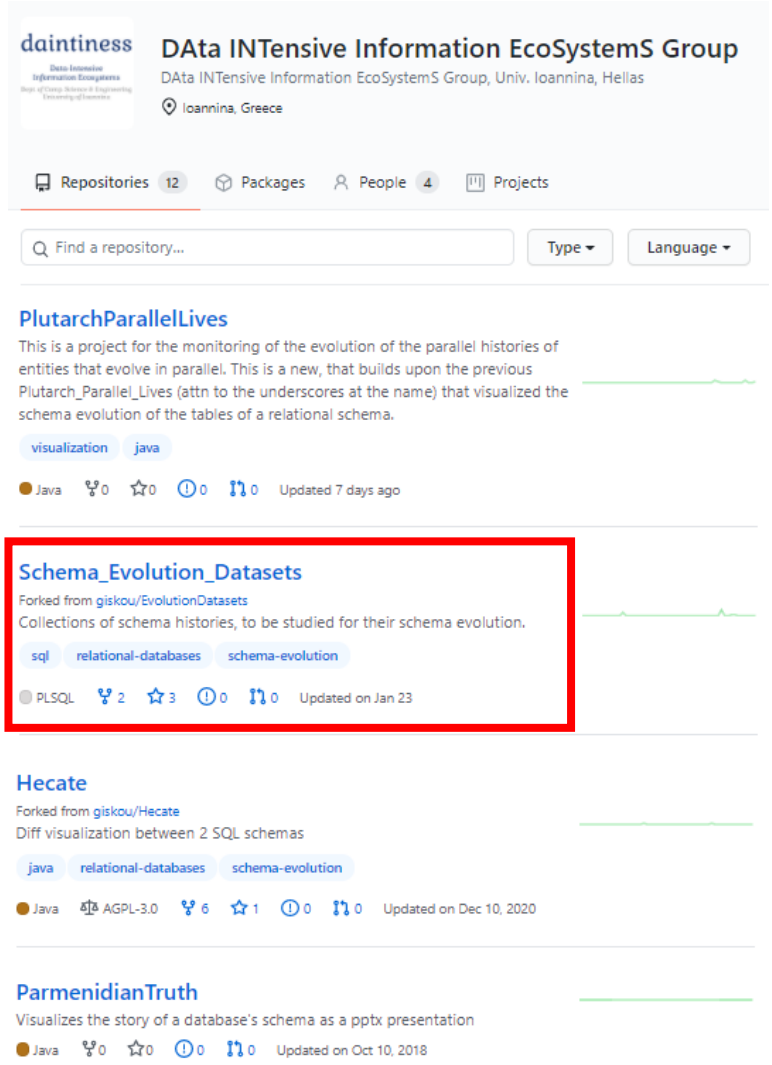
and Code

… for computing differences (Hecate)

… visualizing schema lives (Plutarch Par. Lives)

… visualizing the structure of FK's (Parmenidian Truth)

… handling the impact of evolution (Hecataeus)

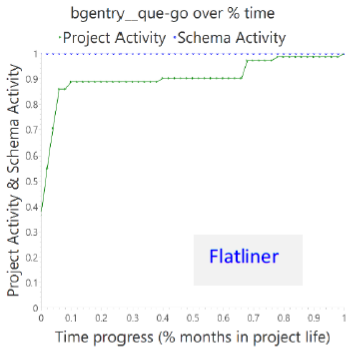


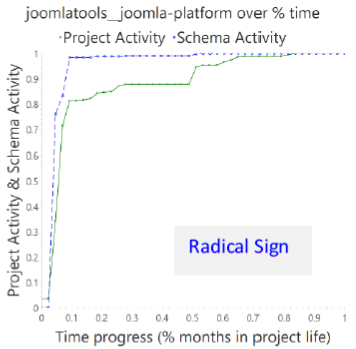
To probe further (code, data, details, presentations, …)

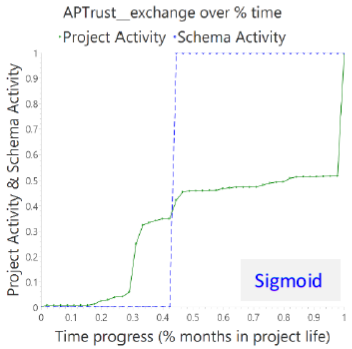
www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

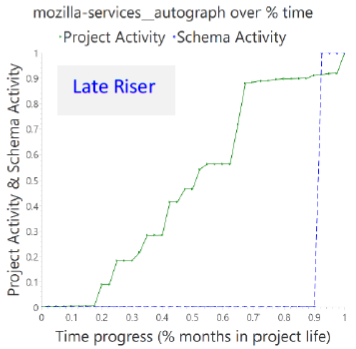
Our core contribution in EDBT 25

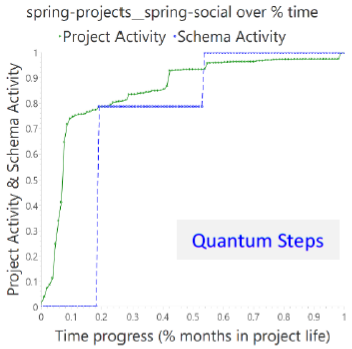
For the first time in the related literature, a set of **patterns on how schemata evolve over time** has been discovered

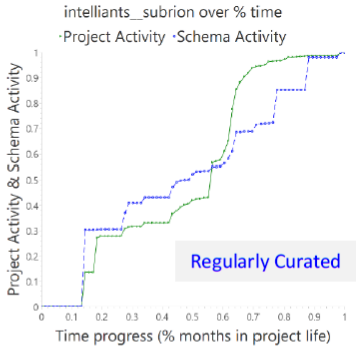


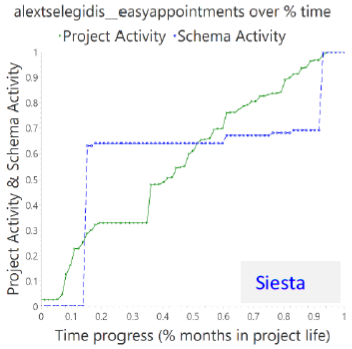


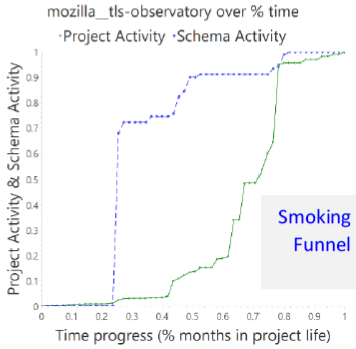










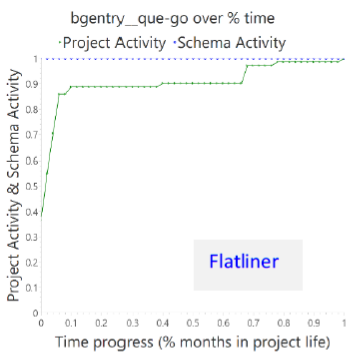


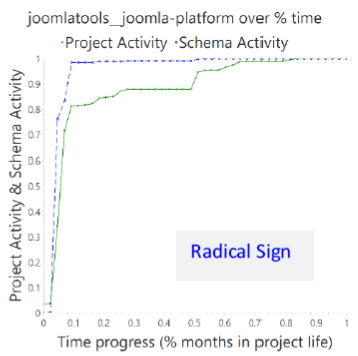
Time-Related Patterns of Schema Evolution

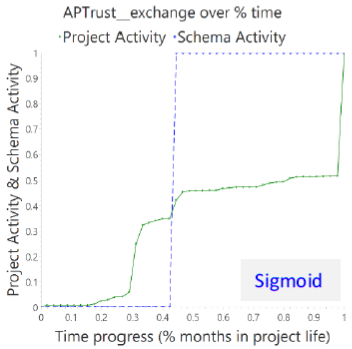
- We extracted…
- … 8 Patterns of Change in Time…
- … organized in 3 Families

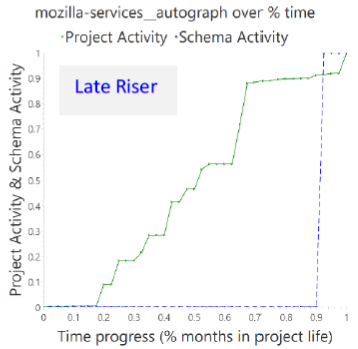
- We verified the results of pattern extraction wrt common sense, generalization, disjointness, cohesion, completeness

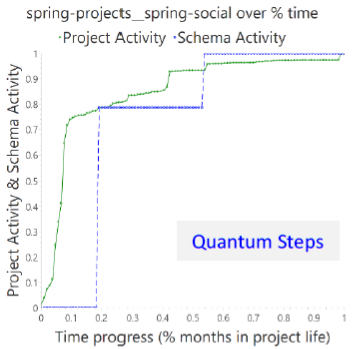
- We related the patterns to other properties of schema change

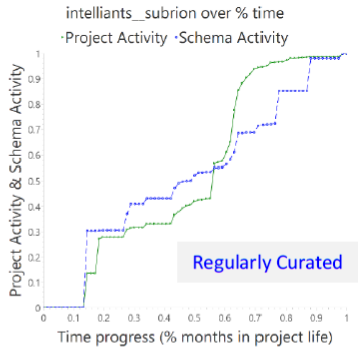


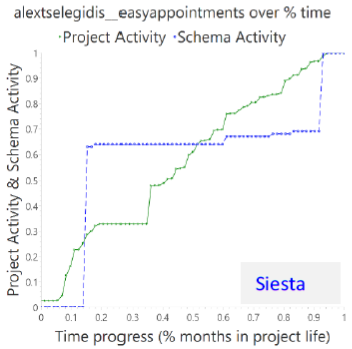


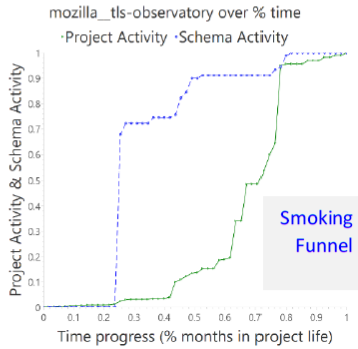




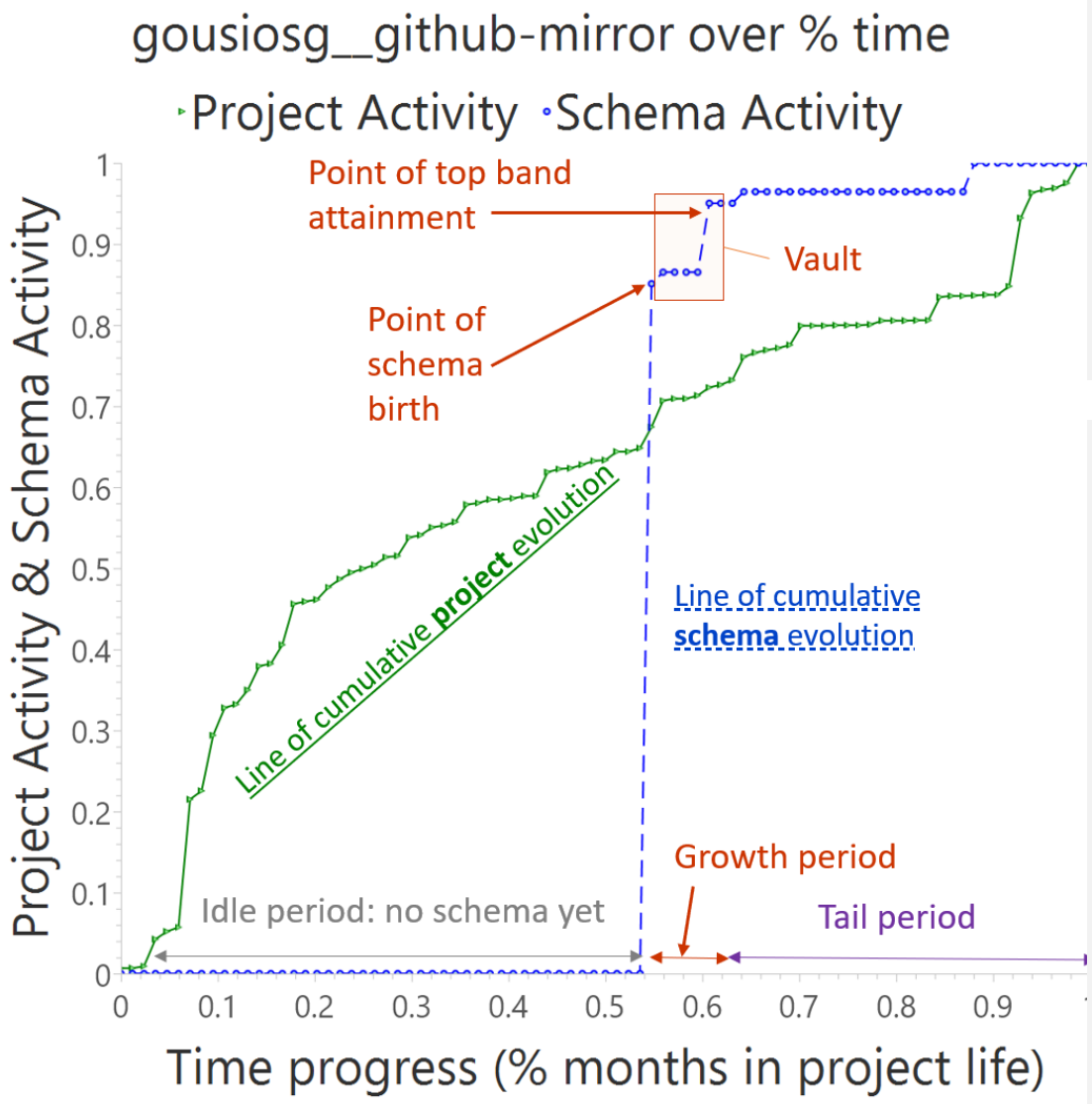








Nomenclature



Schema Update Period: time span between 0th (originating v.) and last commit for schema updates

Project Update Period: resp., for all project updates

Schema Expansion: attr's born with new table, injected to existing tables

Schema Maintenance: att's deleted with deleted table, ejected from surviving table, data type change, PK change

Schema activity = Expansion + Maintenance

Unit of measurement: #affected attributes

Horizontal axis: **time as a percentage of a project's life.**

Vertical axis: **cumulative progress** as a percentage of the total amount of evolution activity, for

(a) the schema (dotted, blue line)

(b) the source code (solid, green line).

Top-band: 90% of total activity

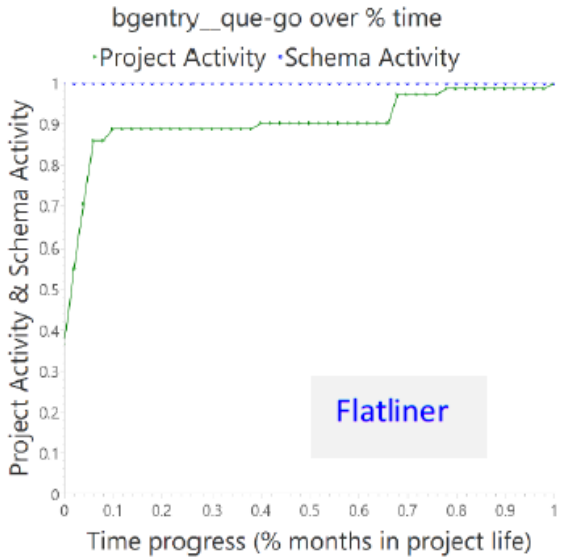
Growth period: between schema birth and attainment of top-band

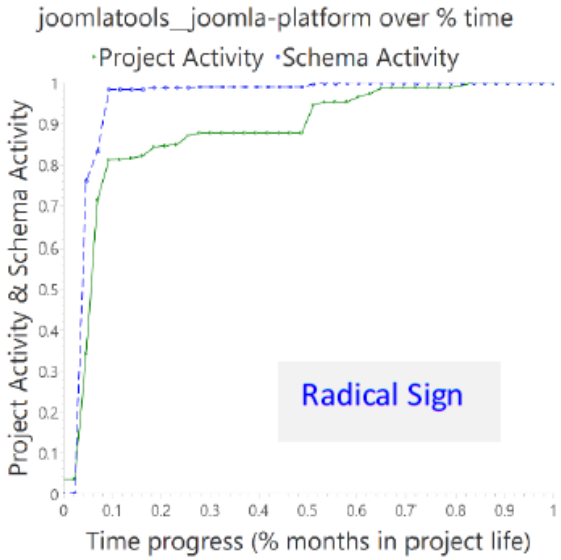
Vault: when the transition between schema birth and top-band takes less than 10% of the total time.

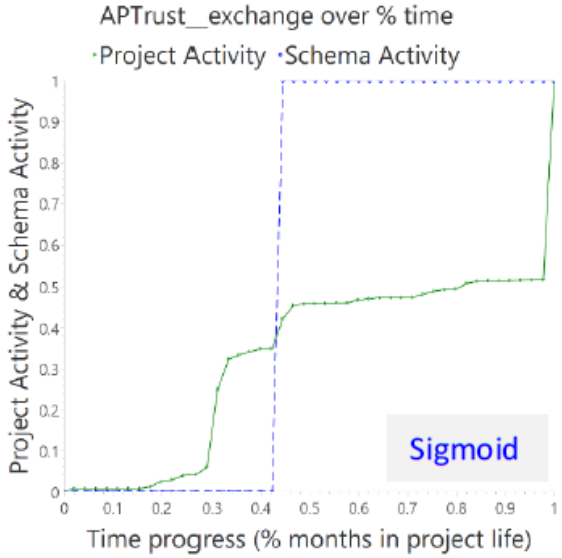
The “**Be Quick or Be Dead**” family of patterns constitutes a family of **very focused change very close to the point of schema birth** - the only difference of the involved patterns is when schema birth takes place.

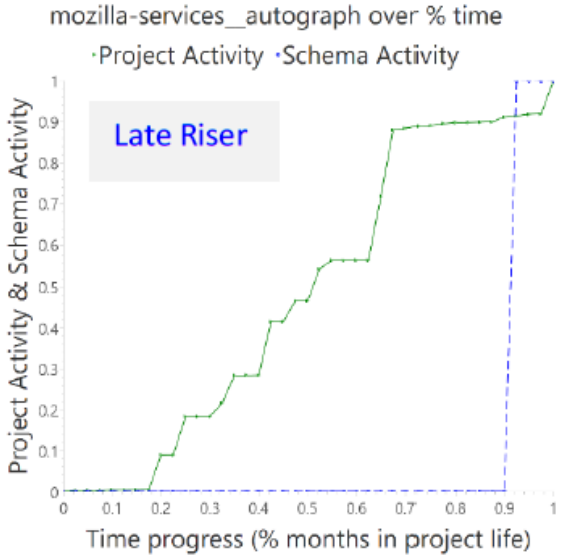
64% prj’s in the corpus

Main characteristic: **aversion to change**







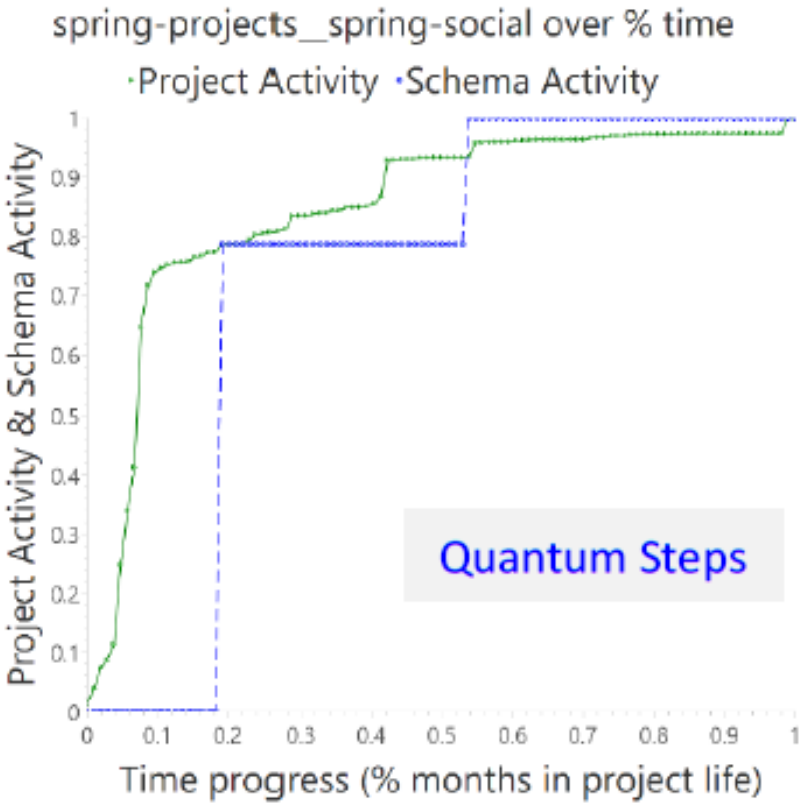


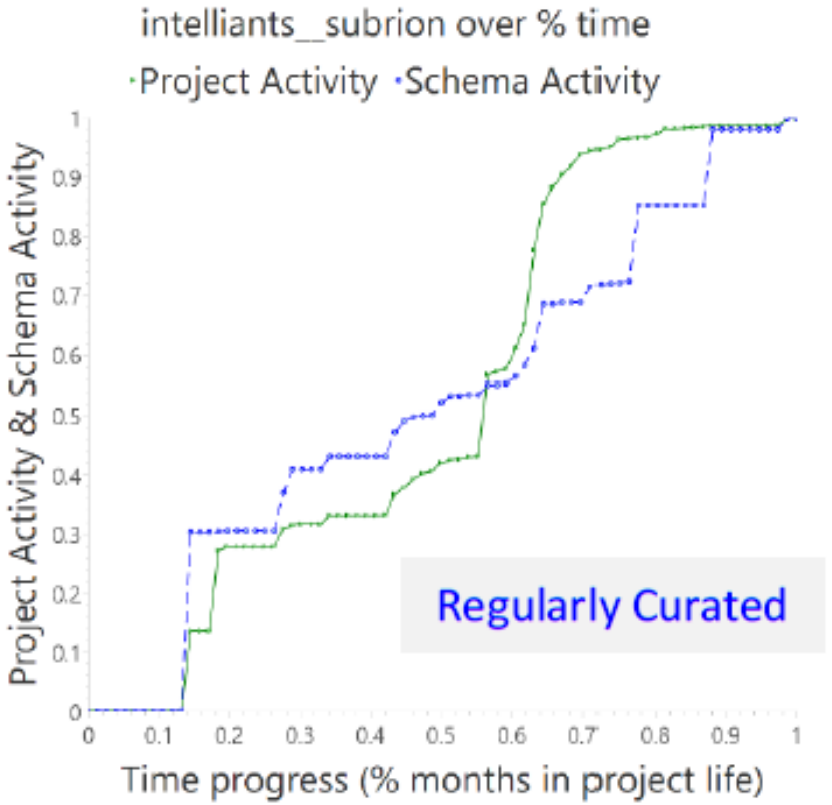
The "**Stairway to Heaven**" family of patterns: both patterns, involve **a fairly regular pattern of change**, with change steps distributed across time.

Although **different in the change rate**, both patterns refer to projects that do not reach the top band in a single shot, but **progressively climb to the top-band over a long period of time**.

Almost 25% prj's in the corpus

Main characteristic: **distribution of change** volume over time

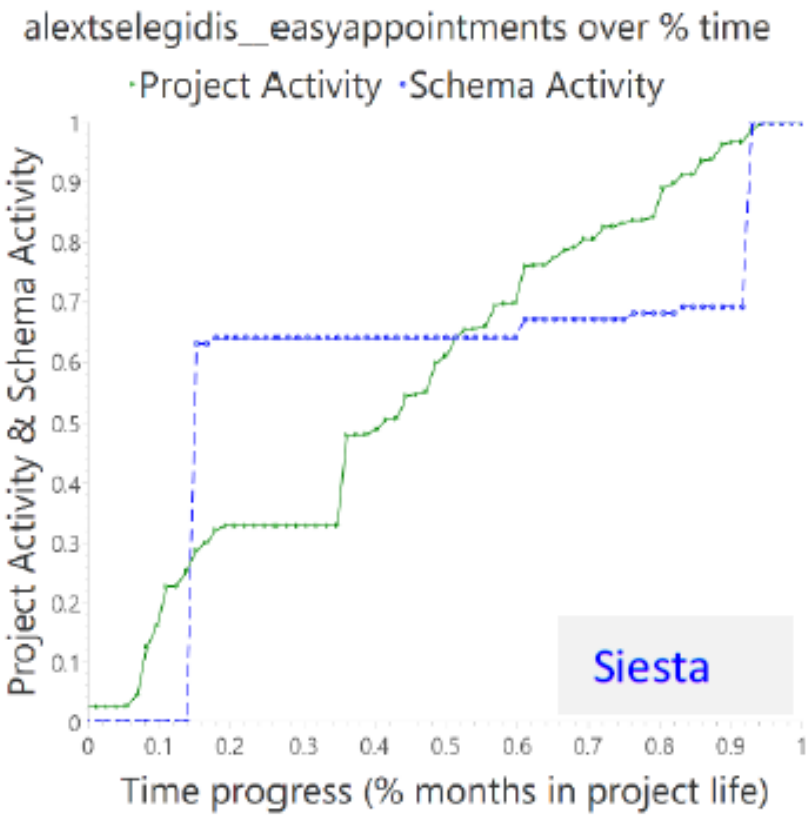


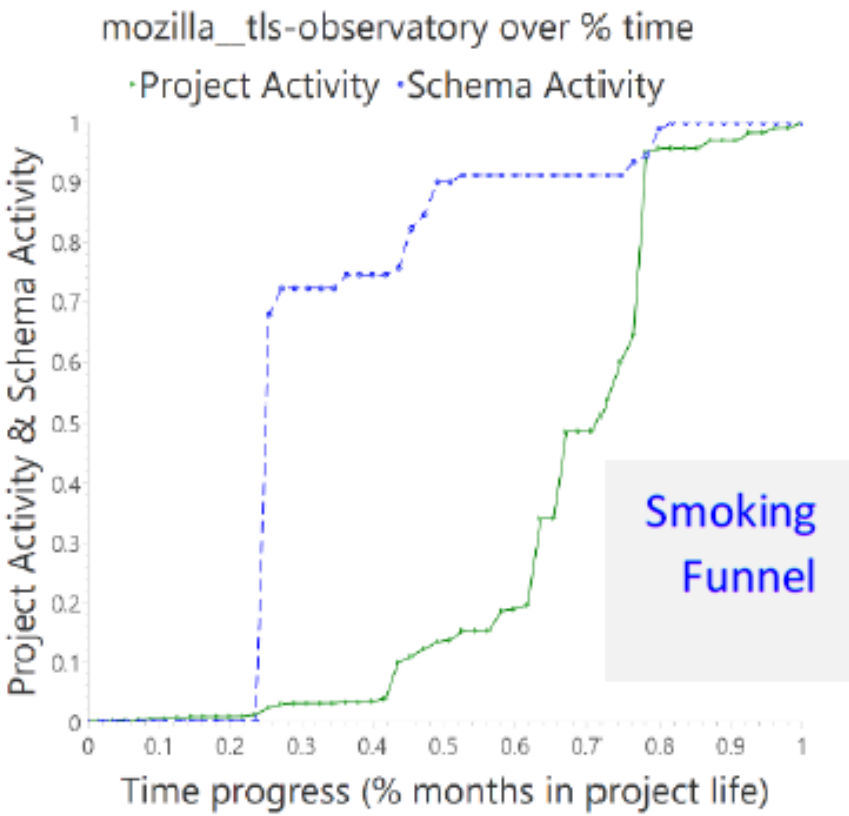


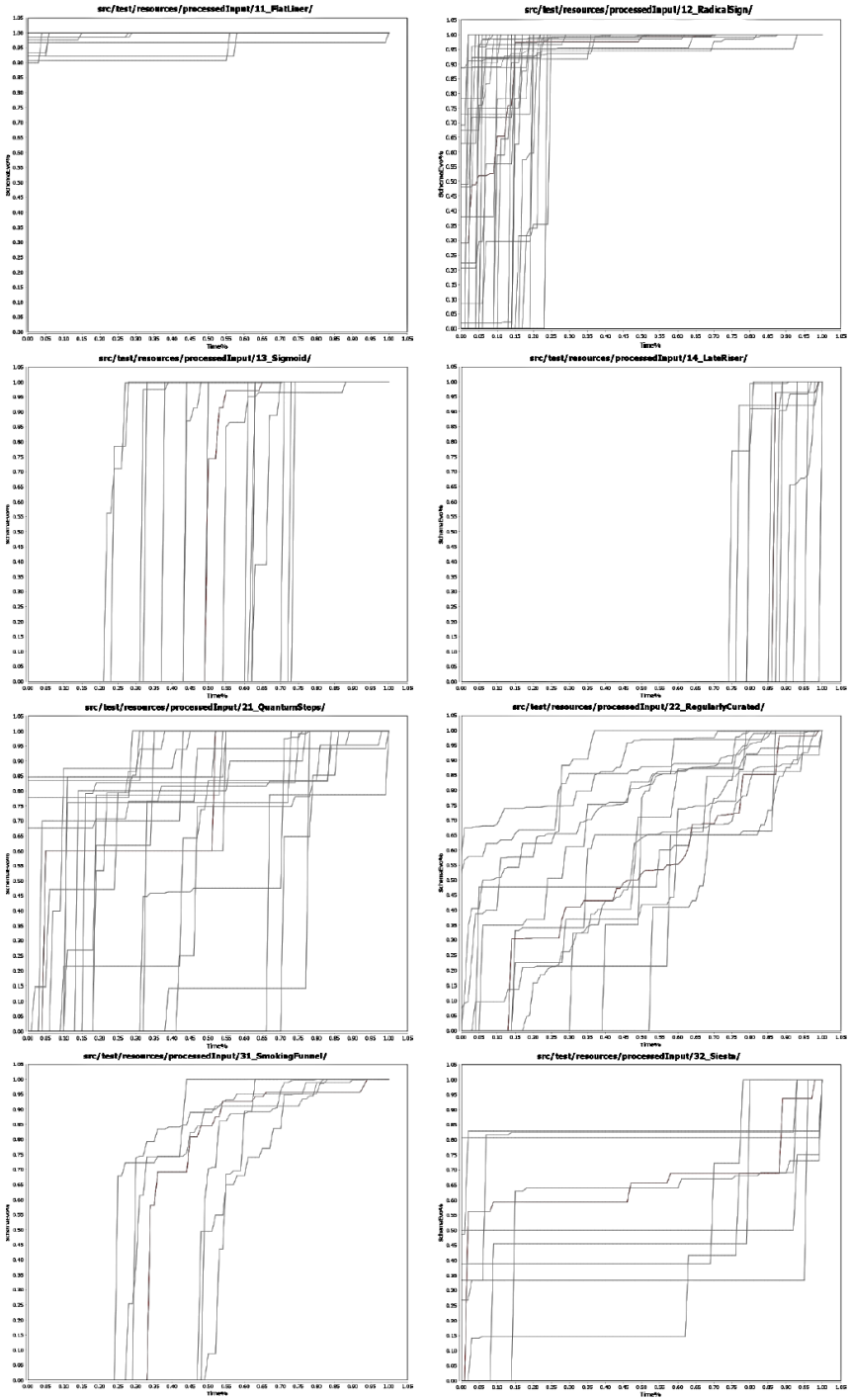
The "**Scared to Fall Asleep Again**" family of patterns: the two patterns, although very different in their characteristics, resemble in that they **include projects where the change is not focused in a single point**, and **happens towards the end of the lifetime of the project**.

11% prj's in the corpus

Main characteristic: **late change**







Validation of patterns

We have positive answers to the following questions:

VQ1: Are these patterns **genuine** and **reasonable**? How can we guarantee that the separation is not artificial and a-posteriori fitted to the numbers?

VQ2: Can we claim that the classification of projects into different patterns is producing patterns that are (a) **internally cohesive** and (b) **pairwise disjoint**?

VQ3: How **generalizable**, i.e., **how representative of the general behavior of projects, are the results?**

VQ4: Is the taxonomy produced **complete**? How possible is it that other behaviors do exist too?

Plz. check out the paper.

Contributions 1/2

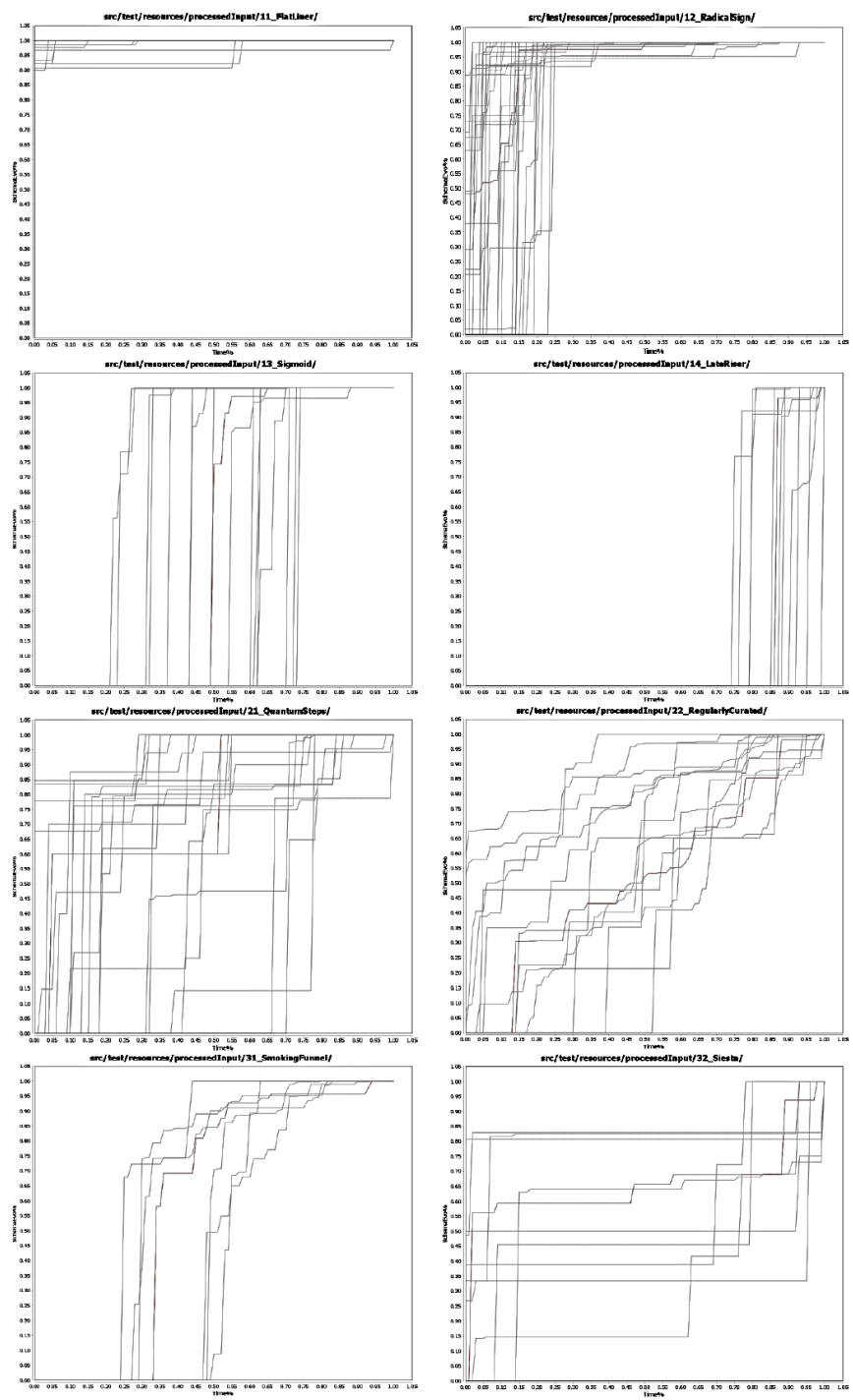
- The core contribution of this paper is the identification of **8 patterns** of schema change in time, organized in **3 families**, The patterns essentially reflect a model of how change is done via **two important traits**:
  - **aversion to change**, practically 2/3 of the corpus, and,
  - **observable, regular evolution**, in several fashions: rare or dense, yet regular, change (amassing to 25% of the corpus), and, surprisingly, an 11% of the corpus with late change too.
- We **verified** the results of pattern extraction **common sense**, **generalization**, **disjointness**, **cohesion**, and **completeness**

Contributions 2/2

- **Other measures of evolution**: Although all patterns have similar PUP, the Smoking Funnel and Regularly Curated projects start bigger and contain larger schema evolution activity than the rest of the projects who start small and typically show lower values of change.
- **Change types**: The projects of the change-averted patterns come with small change, frequently being zero, and an inclination towards expansion. The rest of the patterns come with higher volumes of change, and a variety of change types, mostly towards expansion. Both expansion and maintenance are performed with the granule of change being mostly the entire table.
- **Point of Birth**: 34% of the schemata are born in M0, 60% in the first 6 months and 68% in the first 12 months
- **Prediction**: The point of schema birth, gives an early, coarse indication of the subsequent evolution: if born in M0 or after the first year, the schema has a strong inclination towards rigidity (75% and 64% resp.); birth within the first year however, gives a 53% probability, respectively.

With an eye to the future

- Methodologically, the paper opens a road for future research on **other kinds of schemata** (e.g., do the same kind f research for Nosql schemata)
- Solid foundations for the **prediction of future behavior** on the basis of a meaningful model.
- Obtain and study schema **histories from proprietary schemata** (for the last 50 years of the database discipline, this has been practically impossible).
- Start looking for **patterns and antipatterns** in the design and coupling of schemata and software
- (Even more importantly) **Developing educational material and practical exercises for our students**, in order to train them on the practical aspects of the topic is another important road for the future.



Thank you!

EDBT ICDT BARCELONA 2025

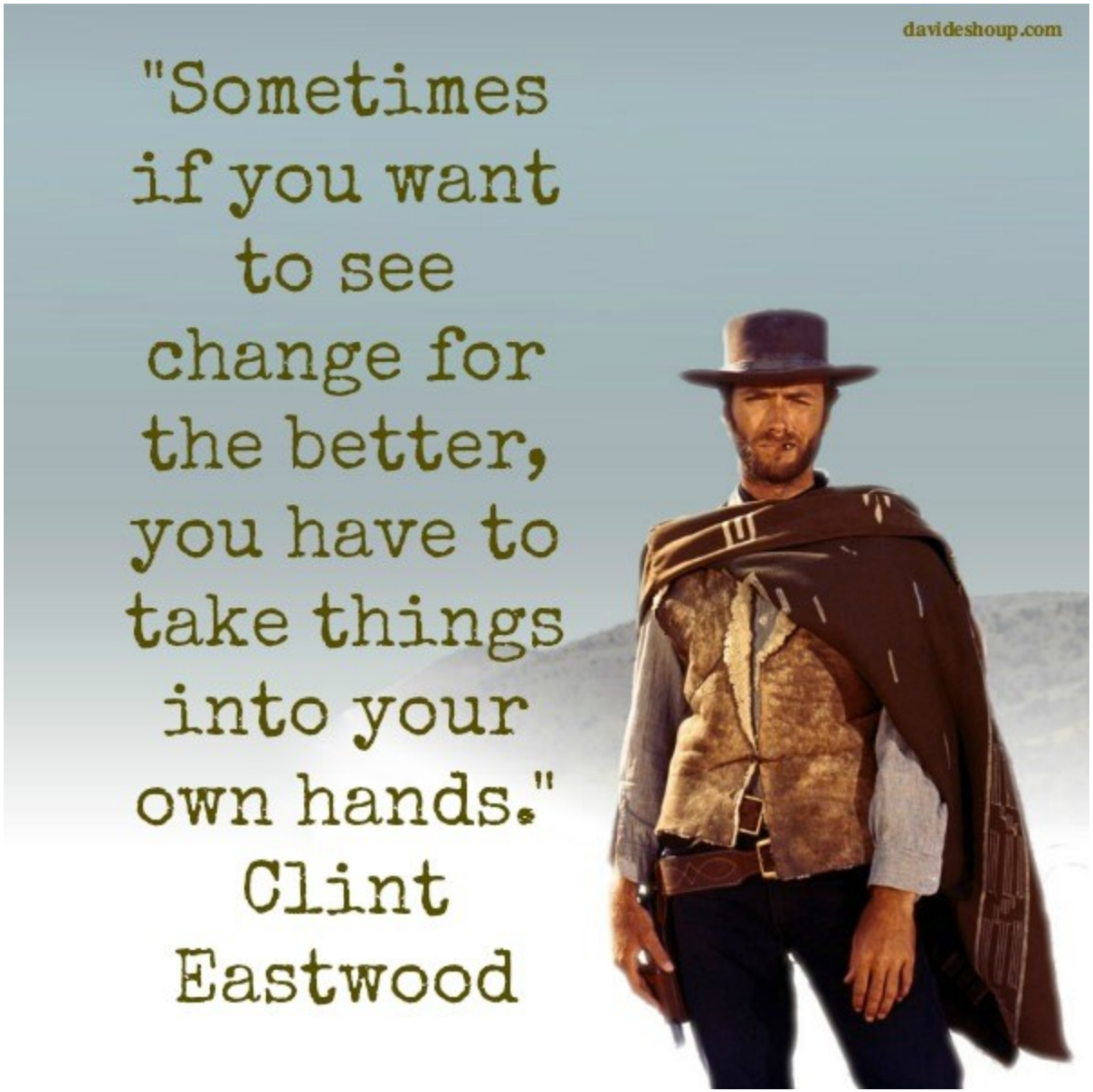
With many thanks to our organizers!

To probe further (code, data, details, presentations, …)
www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

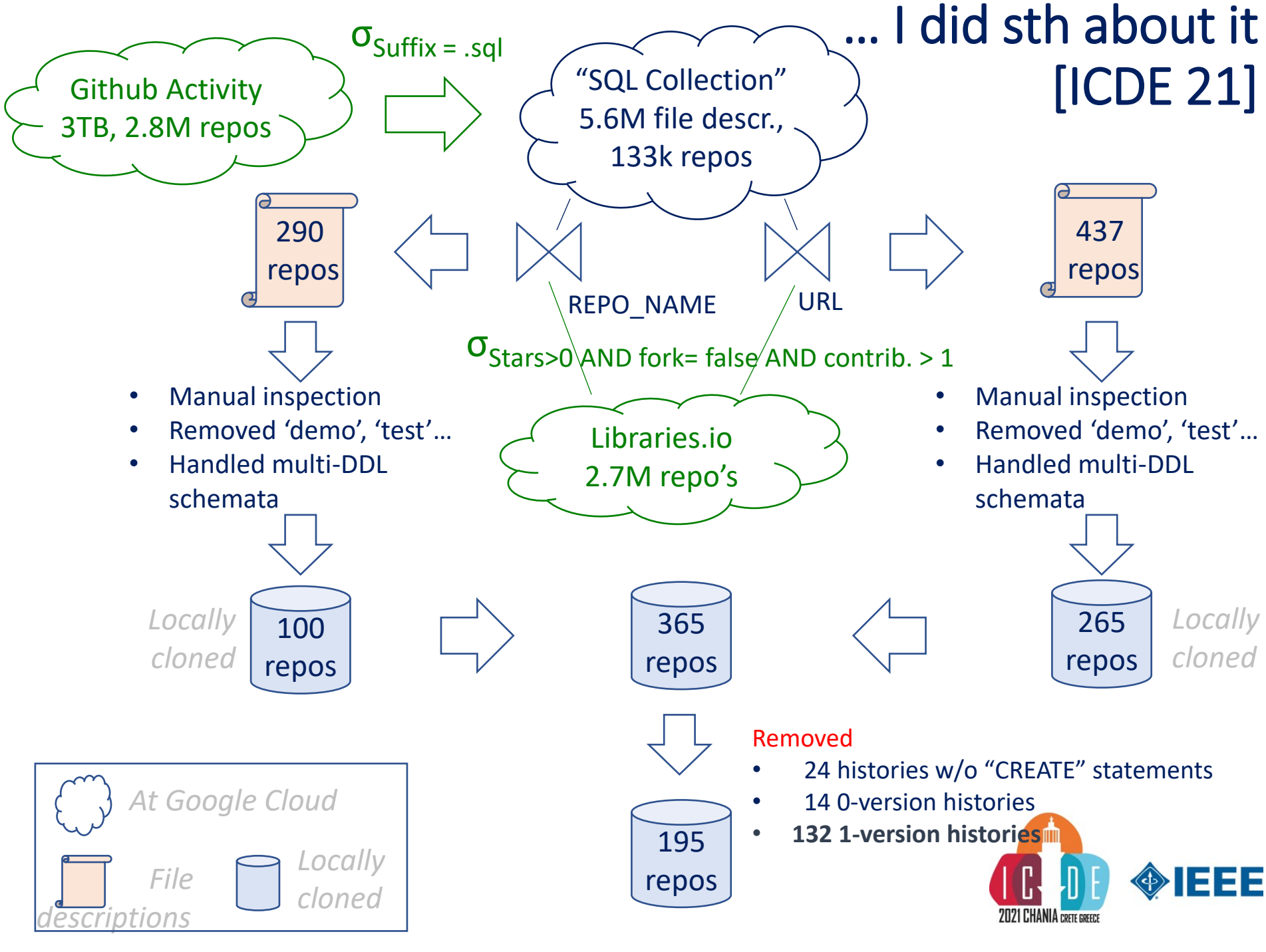
Data and code
https://github.com/DAINTINESS-Group/

… neither do schemata (most times, but not always) …

Auxiliary slides cache


davideshoup.com
"Sometimes if you want to see change for the better, you have to take things into your own hands."
Clint Eastwood

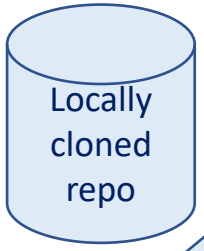
… I did sth about it [ICDE 21]



We work with significant projects

- In whatever follows, remember that we have not selected just any random project, but rather,…
- we intentionally restricted our scope to original, stared projects, where people were actually contributing effort to develop and maintain.
- Overall, 65% of projects spanned more than 24 months and 77% more than a year.

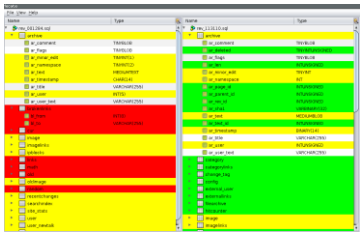
Post-identification workflow for each of the 195 projects



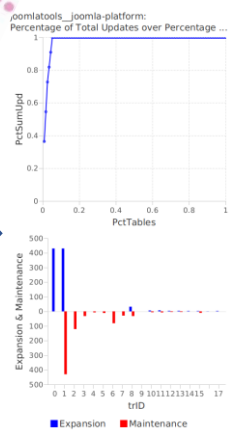
Project's cloned repo from GitHub

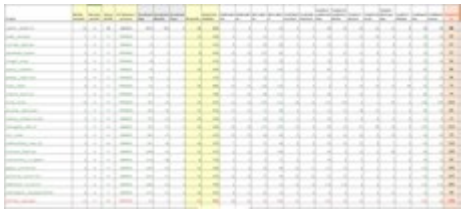
Schema History = sequence of DDL committed snapshots

Hecate: SQL schema diff extractor



Heraclitus Fire: a chart/stats extractor





https://github.com/DAINTINESS-Group/Hecate

https://github.com/pvassil/HeraclitusFire

https://github.com/DAINTINESS-Group/Schema_Evolution_Datasets/

Scope of the study

- We are interested in the monitoring of the **evolution** of the logical-level relational **schema** for significant Free Open Source Software projects, hosted in GitHub.
- We are not covering or generalizing to
  - … proprietary schemata outside the FoSS domain,
  - … conceptual or physical schemata,
  - … non-relational schemata, e.g., XML, JSON, …

Pattern Extraction Methodology

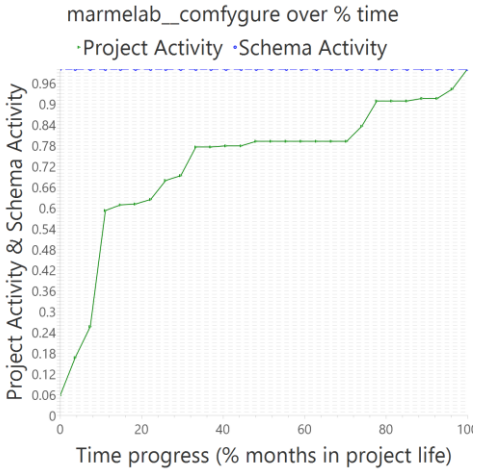
1. We have excluded all projects with a life time less or equal to 12 months: 151 projects.
2. We **manually** searched for patterns of the schema line and annotated projects accordingly.
   - This process was iterative, in several rounds and based solely on the aforementioned visual representation of the cumulative progress of schema evolution.
   - Why intentionally manual? Typical in research design s.t. a golden standard of meaningful, humanly-verified groups is attained first, and then checked on the rest of the properties
   - See the paper for pointers on **Grounded Theory** for iteratively extracting patterns out of data
3. **Quantitatively verified the disjointness ,cohesion and completeness** of the patterns and grouped patterns in larger families.
4. Quantitatively analyzed how patterns related to other properties of schema evolution

All data, results, charts and auxiliary analyses are available at :

https://github.com/DAINTINESS-Group/Schema_Evolution_Datasets/

Flatliners

Everything happens at birth

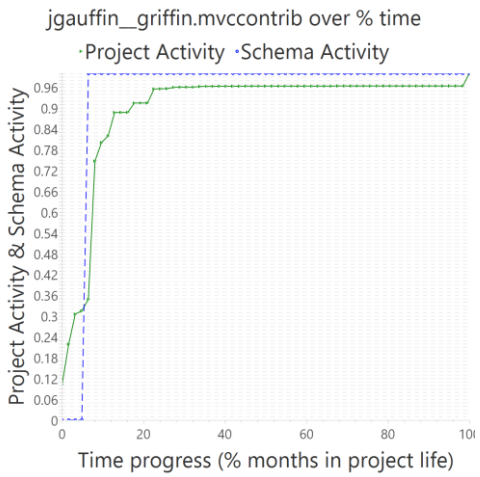


#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
23	**V0** Born really early…	**V0** … via a single vault…	Zero …that does 100% of the job	Full	0

Exceptions: -

Radical Sign

Born early, very soon freezes totally



#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
41	**V0, Early** Born early …	**Early** …mostly via a single vault;	zero, soon, fair Trip to top is fairly short;	Long All ends soon, i.e., a long tail	0 - 2

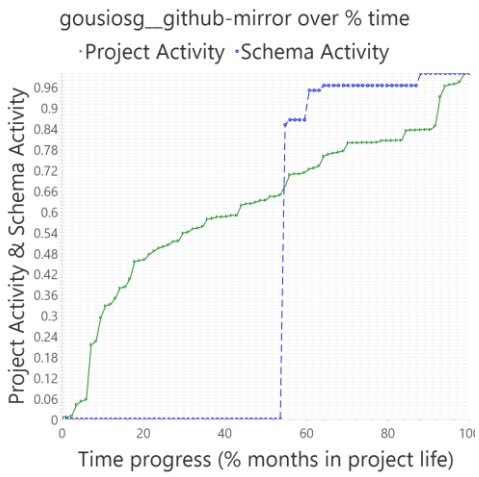
Exceptions: -

Sigmoid

Born in the middle, sharp vault to top-band

Resembles a "pure" sigmoid function, better than any other pattern

(almost all patterns are Exceptions of a sigmoid)

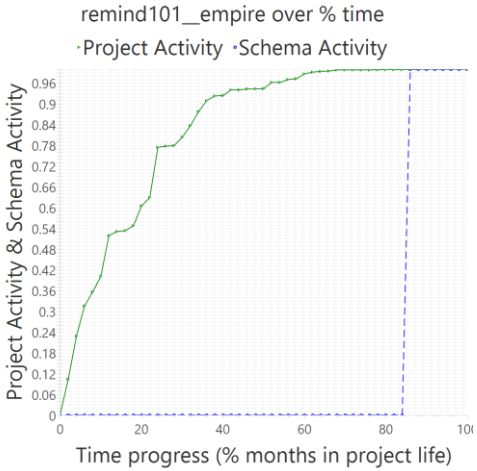


#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
19	**Middle** Single vault in the middle	**Middle**	**Zero, Soon** (~immediately, with a single vault)	Fair	0 - 1

Exceptions: (2 exceptions, early born)

Late Riser

Born late, sharp vault to top-band



#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
14	**Late**	**Late**	**Zero, soon** No early or middle life, late single vault	Short	0
Exceptions:		(1 exc.: middle)			(1 exc.: 5 months)

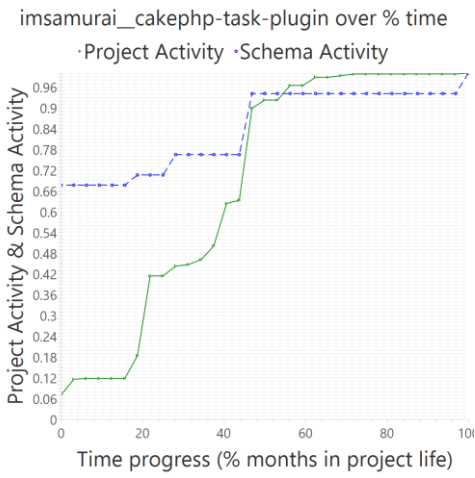
Quantum Steps

A sequence of a few focused steps in the middle…

Two variants, both with few updates:

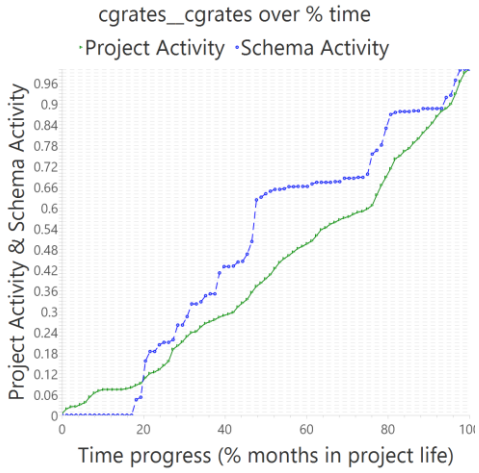
(a) early start, middle top, and,

(b) middle start, late top



#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
17	**V0 or Early**	**Middle**	Fair or Long	Fair	**0 – 3**
6	**Middle**	**Late**	Fair or Long	Soon	**0 – 3**
23					
Exceptions:		2 exc.			

Regularly Curated



Grows “regularly” over time with activity.

Reaches top, with activity, in two variants:

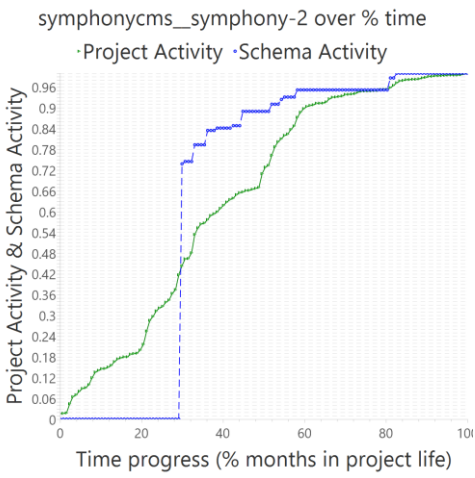
- If born early, reaches top middle or late;
- If born middle, reaches top late

Schema line close to the green line of prj evo, occ. with a small tail

#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
11	**V0 or Early**	**Middle or Late**	Long or very long	Soon	**> 3**
3	**Middle**	**Late**	Fair or long	Soon	**> 3**
14					
Exceptions:	-				

Smoking Funnel

Somewhat late birth, with something like a vault (but not full or super high), and once born, alive with regular schema updates



#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
7	**Middle**	**Middle**	**Fair**	Fair	> 3
	Born in the middle …	… and reaches top in the middle…	… but with some action in the way…		… thus the activity

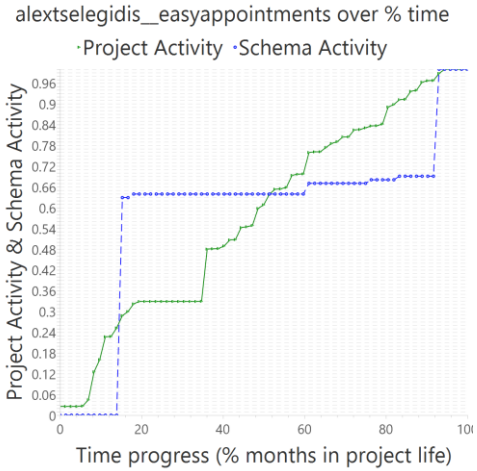
Exceptions: -

Siesta

Born early, at a moderate high level,

then goes to sleep for some (significant) time,

then wakes up again



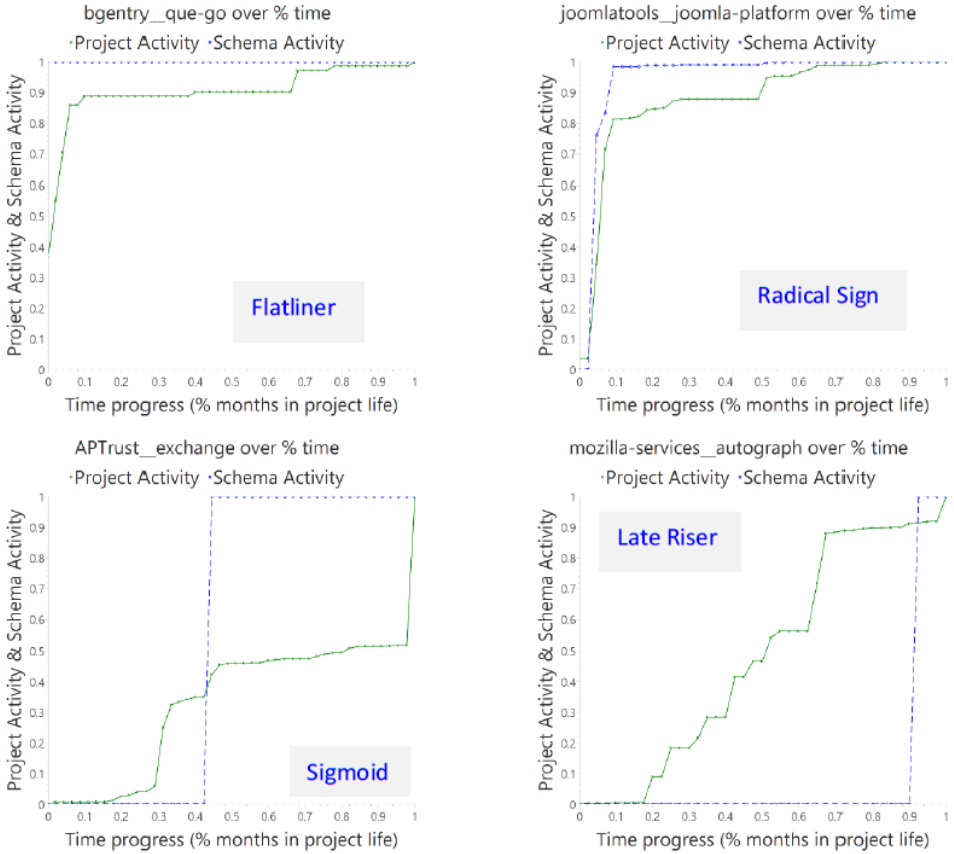
#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% very long > 75% time*	*How long from reach-of-topBand to end? Soon <=25%, fair (25% .. 75%], long (75% .. 100%) Full 100%*	*Growth: [birth-topBand)*
10	**V0 or Early**	**Late**	**Very Long**	Soon	**0 – 3**
	Early born…		…with a long sleep in the middle..	… and some action in the end	i.e., without much ado
Exceptions:			(1 exc.: long)		(2 exc.)

Traits of schema evolution

- The patterns reflect essentially how change is done, rather than just being statistically-backed project clusters. We observe two important traits:
- The first trait is the aversion to change, aka progressive gravitation to rigidity, meaning that curators avoid change as much as they can, and more often that not, the schema freezes after a few changes. This is a majoritarian trait, and concerns the Be quick or Be Dead family, which involves practically 2/3 of the corpus.
- The second trait concerns a minority of projects whose curation team regularly synchronizes the schema to the surrounding changes, with observable regular schema evolution, coming in several fashions: rare but regular change; densely regular change; and, surprisingly, late change too.
- Overall:
  - the anecdotal evidence of “freeze the schema first; then build all the applications on top of it”, although certainly majoritarian as a practice, is only partially corroborated, with the existence of projects that are maintained “regularly” in various fashions
  - The assertion of several works in the related work, that change is frequent, is also mostly disproved – the reason is that, out of necessity, if change is to be studied, it has to be studied

Aversion to change

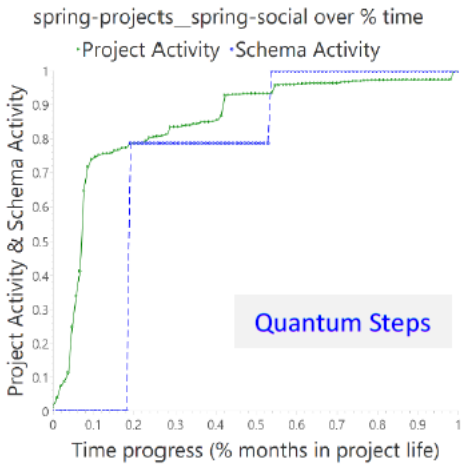
- Almost 2/3 of the corpus, 97/151 evolve with very focused change very close to the point of schema birth - the only difference of the involved patterns is when schema birth takes place.
- This is **in sync with our recent previous findings over a very large corpus of projects**, where 70% of 327 projects investigated showed very little – if any- signs of change.

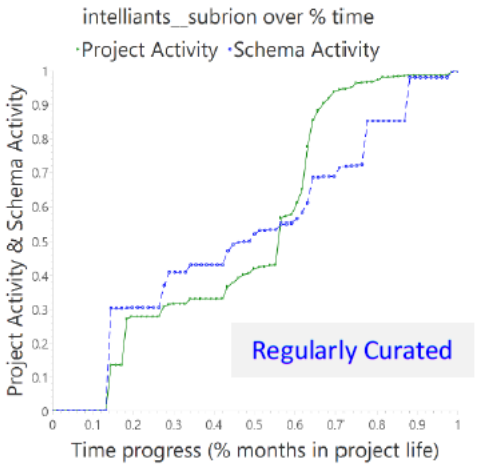


- This is in contrast with all the past research till the 20's, involving small studies, of few, carefully picked prj's that actually showed any change whatsoever

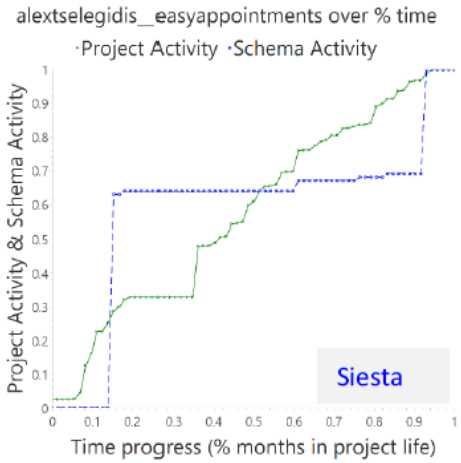
Still, change exists!

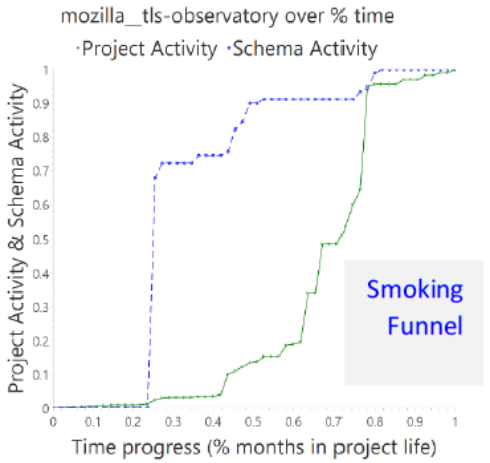
- Almost 25% (37/151) projects evolve with regular change steps, distributed across time, progressively climbing to the top-band over a long period of time.





- There is even late change! 17/151 projects come with change not focused in a single point, happening towards the end of the lifetime of the project.





Summary of simple stats

- **Birth is mostly done early**
  - Two thirds of the projects (105 projects) see schema birth at V0 or before 25% of the PUP.
  - 74 schemata (half the corpus) are born in the first 10% of time.
- **Change is mostly sharp in time:**
  - 115 projects, 76% of the population had no more than 1 month of activity from schema birth to top.
  - 88 / 151 projects (58%) had a single vault, i.e., an interval from schema birth to top-band that was less than 10% of the PUP, with 62 of them in zero time.

What's the chance of the schema freezing depending on when it was born?

Born in …	%corpus	% chance to end up as be quick or be dead	% chance to end up with regular change (kinda)
M0	34%	75%	25%
[M01 – M06]	25%	53%	47%
[M07 – M12]	9%	53%	47%
>M12	32%	64%	36%

- 60% born in the first 6 months
- 68% in the first 12 months

	OVERALL		Born M0		Born [M1.. M6]		Born [M7..M12]		Not Born till M12	
	#prj's	prob (%)	#prj's	prob (%)	#prj's	prob (%)	#prj's	prob (%)	#prj's	prob (%)
Flatliners	23	15%	23	44%						
RadicalSign	41	27%	16	31%	19	50%	5	38%	1	2%
Sigmoid	19	13%			1	3%	2	15%	16	33%
Late Risers	14	9%							14	29%
Quantum Steps	23	15%	4	8%	11	29%	2	15%	6	13%
Regularly Curated	14	9%	3	6%	4	11%	3	23%	4	8%
Smoking Funnel	7	5%							7	15%
Siesta	10	7%	6	12%	3	8%	1	8%		
	151	100%	52	100%	38	100%	13	100%	48	100%