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

What are the laws of database schema evolution?

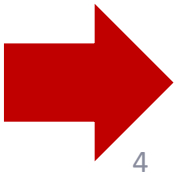
What are the laws of database (schema) evolution?

- How do databases change?
- In particular, how does the schema of a database evolve over time?

- Long term research goals:
  - Are there any "invariant properties" (e.g., patterns of repeating behavior) on the way database (schemata) change?
  - Is there a theory / model to explain them?

Why care for the laws/patterns of schema evolution?

- Scientific curiosity!
- Practical Impact: DB's are dependency magnets. Applications have to conform to the structure of the db…
  - typically, development waits till the "db backbone" is stable and applications are build on top of it
  - slight changes to the structure of a db can cause several (parts of) different applications to crash, causing the need for emergency repairing



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*"

FOSS is an opportunity

- 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:
  - not only is the code available, but also,
  - public repositories (git, svn, …) keep the entire history of revisions
- We are now presented with the opportunity to study the version histories of such "open source databases"

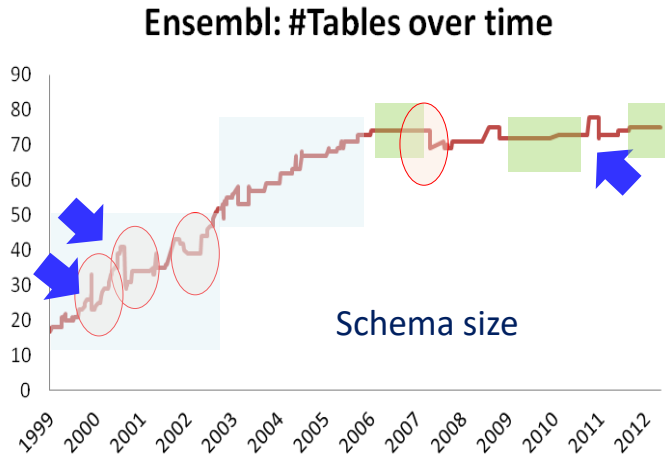
From our ICDE 2021 data set collection process that poked 327 repo's….

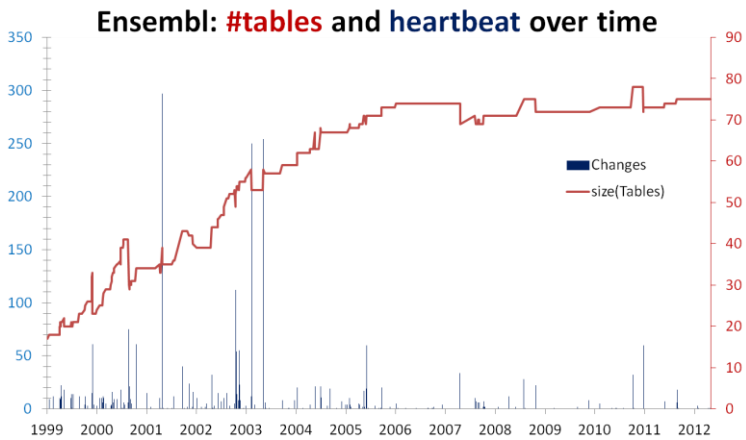
- 70% of the projects, demonstrated total absence or very small presence of change.
- Out of the 327 repositories that we cloned,
  - 132 (40%) had a single commit for their schema (i.e., no change) whatsoever,
  - 34 (10%) had more than 1 commits, but zero changes at the logical-level schema, and,
  - 65 (20%) were almost frozen (with less than 4 active commits and 10 modified attributes).

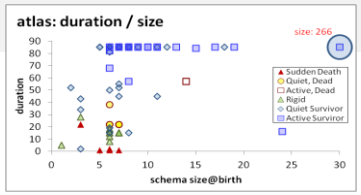
  Within the 195 that we cloned
- We have called this phenomenon **gravitation to rigidity** in our past research [IS15, IS17, JoDS17]

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

Gravitation to Rigidity: the reluctance to evolve the schema is omnipresent, stronger than the tendency to evolve, and grows stronger over time!

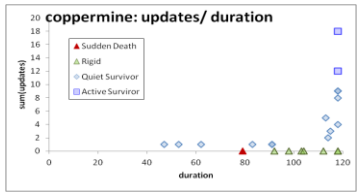




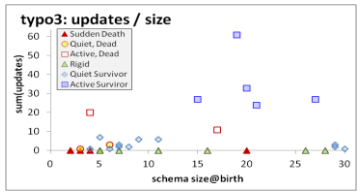


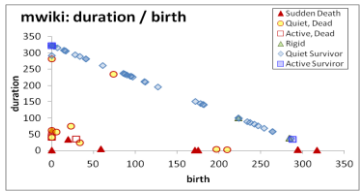
Γ

Individual Tables

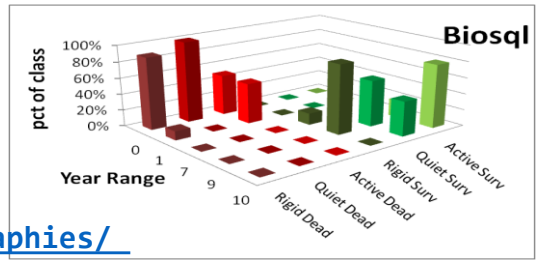


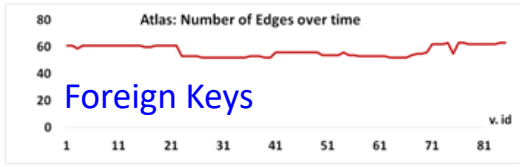
J

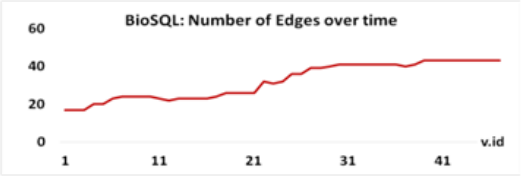


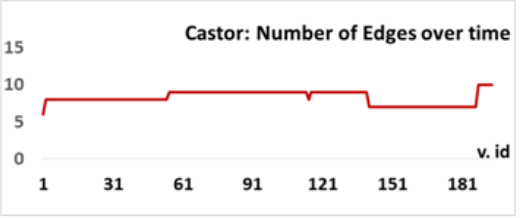


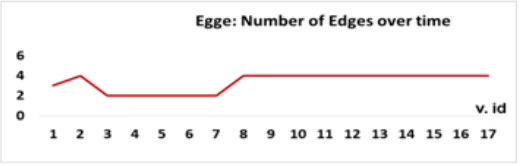
void

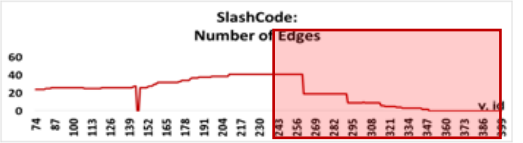


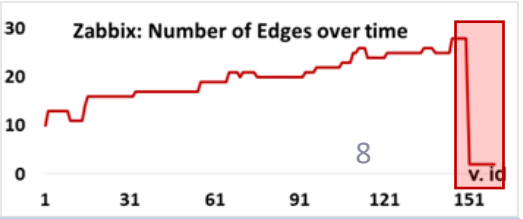












To probe further (code, data, details, presentations, …)
www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

Goal of the paper

- During the last 10 years, we have seen several studies of the characteristics of relational schema evolution with a focus on the volume, change-type breakdown and impact to surrounding queries – however, to the best of our knowledge, none has come up with patterns of how schemata evolve over time.

- The goal of this paper is to investigate the timing nature of schema evolution and extract patterns of how schemata evolve over time.

Our core contribution

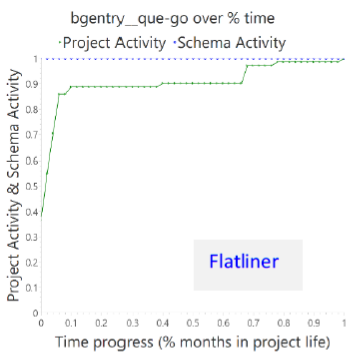
- The major contribution of this paper is that, for the first time in the related literature, a set of time-related patterns on how schema evolution unfolds in time is introduced.
- To achieve this, we had to overcome several limitations:
  - Need to use a large corpus of schema histories to obtain generalizable results
  - Absence of any prior knowledge on how these patterns would look like
  - No established method in our discipline for extracting patterns from schema histories
- To address the volume challenge, we employ a large corpus of 195 relational, logical-level schema histories, out of which we extracted 151 schema histories of length higher than 12 months for further study.
- To address the methodological gap, we have employed methods from empirical software engineering, and iteratively grouped the collected schema histories in patterns of similar time evolution in a qualitative fashion, later to be quantitatively verified, too. We also verified the cohesion and disjointedness of the patterns.

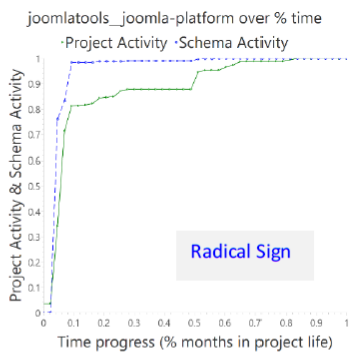
Time-Related Patterns of Schema Evolution

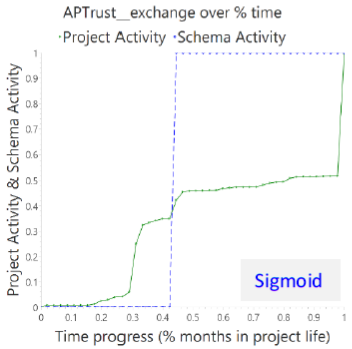
- We extracted…
- … 8 Patterns of Change in Time…
- … organized in 3 Families

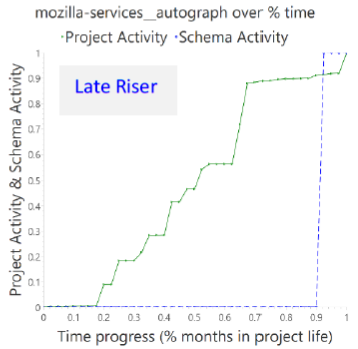
- We verified the results of pattern extraction wrt common sense, generalization, disjointness, cohesion, completeness

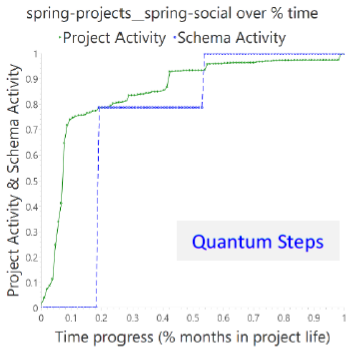
- We related the patterns to other properties of schema change

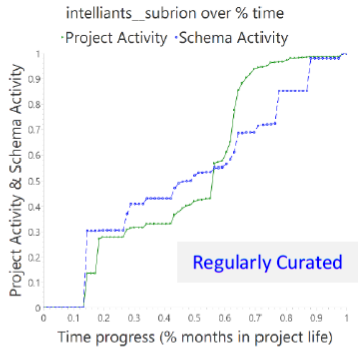


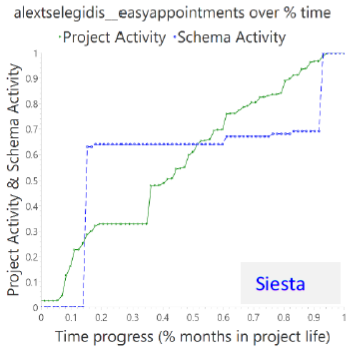


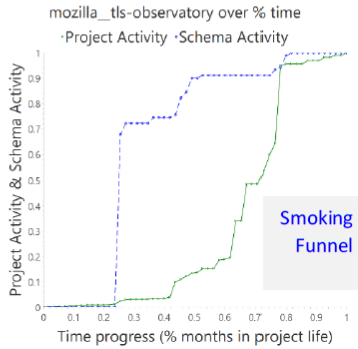












Contributions 1/2

- The core contribution of this paper is the identification of **8 patterns** of schema change, organized in **3 families**, on how developers and data curators regulate schema evolution over time.
- 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.

Experimental Setup and Nomenclature

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, …

Extraction Process: aimed for massive history collection at very large numbers

- **We use the ICDE 2021 Schema Evolution data set (*P. Vassiliadis. Profiles of Schema Evolution in Free Open Source Software Projects. ICDE 2021*).**

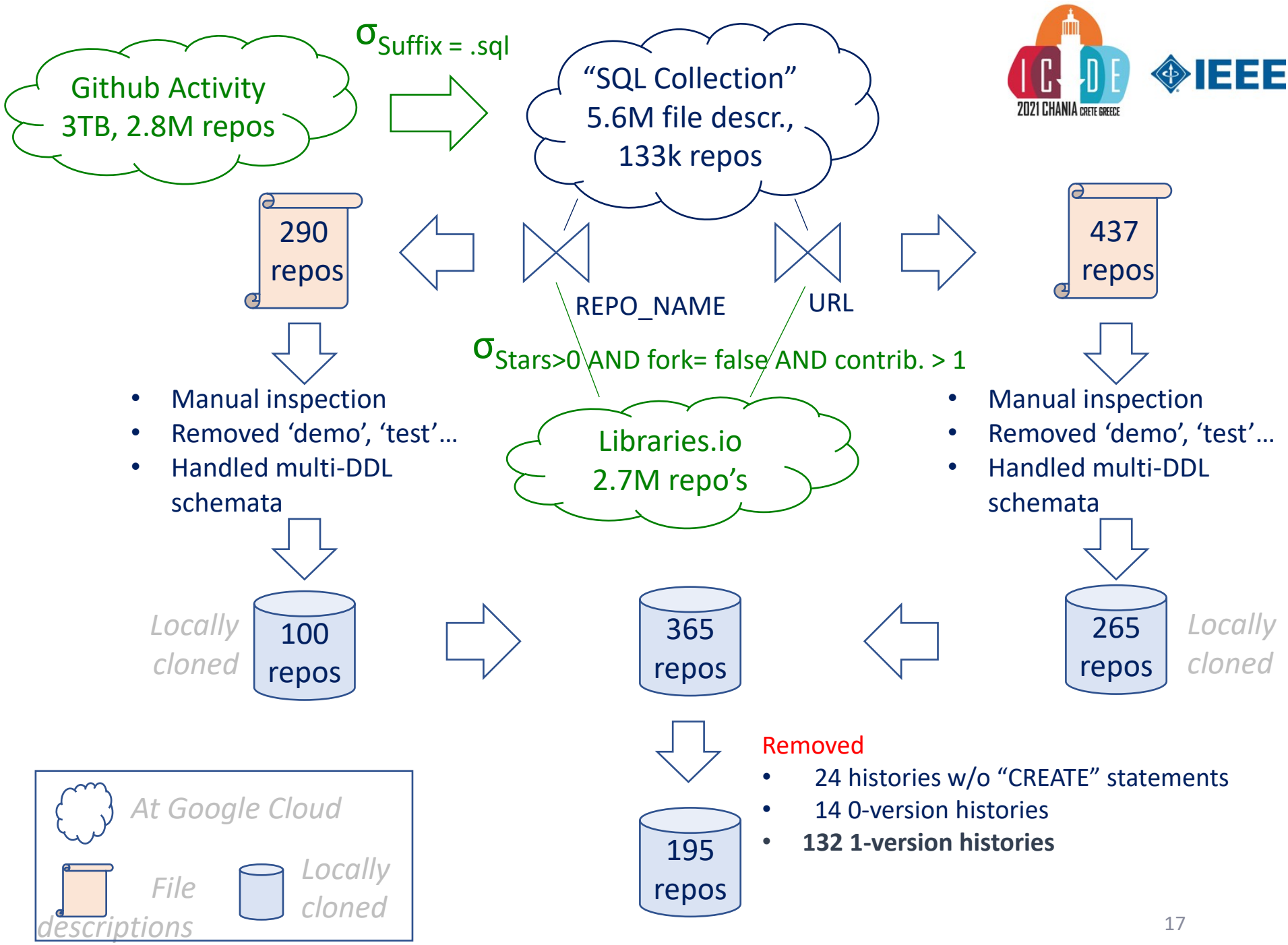
- Queried the GitHub Activity Data dataset from Google Cloud BigQuery (a 3TB+ dataset that contains a full snapshot + the commits of more than 2.8 million open source GitHub repositories) for repos having .sql files → 133K repo's
- Joined this with Libraries.io dataset (metadata for > 2.7M FOSS prj's) and
- filtered for
  - original repositories, with more than 0 stars, more than 1 contributor
- excluding
  - all files with 'test' or 'demo' or 'example' in the path
  - instances of multiple appearances of a DDL file for >1 vendors
  - multiple DDL's (file-per-table mode), incremental maintenance, vendor X language Cartesian Products

=> 365 candidates, locally cloned, cleaned from empty .git, .sql files with no CREATE TABLE statements, …, which eventually led to the final data collection.

=> RESULT: 327 histories out of which

… 132 (40%) with just a single commit ⇔never changed ANYTHING)!!!

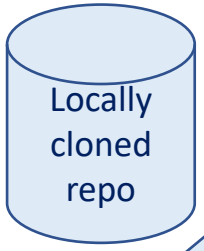
… 195 histories with at least an extra commit, which we subsequently used


Github Activity
3TB, 2.8M repos
σSuffix = .sql
"SQL Collection"
5.6M file descr.,
133k repos
290
repos
437
repos
REPO_NAME
URL
σStars>0 AND fork= false AND contrib. > 1
Libraries.io
2.7M repo's
• Manual inspection
• Removed 'demo', 'test'…
• Handled multi-DDL schemata
• Manual inspection
• Removed 'demo', 'test'…
• Handled multi-DDL schemata
Locally cloned
100
repos
365
repos
265
repos
Locally cloned
195
repos
Removed
• 24 histories w/o "CREATE" statements
• 14 0-version histories
• 132 1-version histories
At Google Cloud
File descriptions
Locally cloned

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.

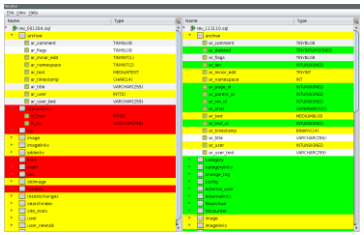
Post-identification workflow for each of the 195 projects



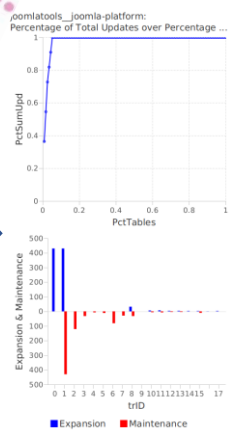
Project's cloned repo from GitHub

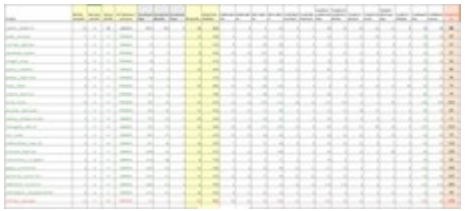
Schema History = sequence of DDL committed snapshots

Hecate: SQL schema diff extractor



Heraclitus Fire: a chart/stats extractor



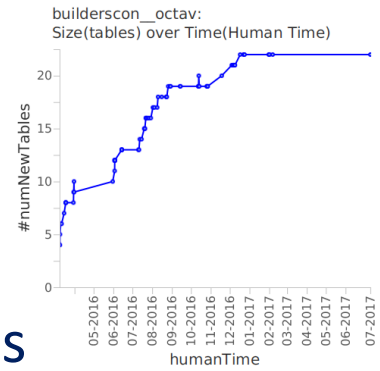


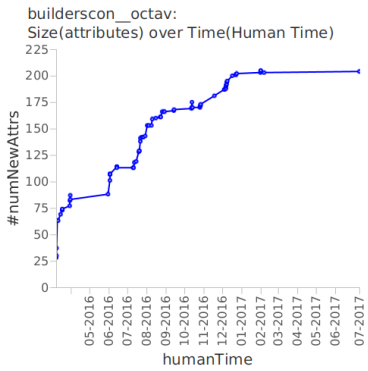
https://github.com/DAINTINESS-Group/Hecate

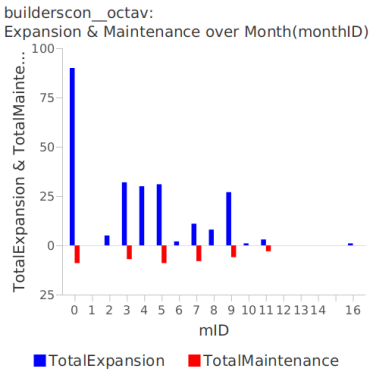
https://github.com/pvassil/HeraclitusFire

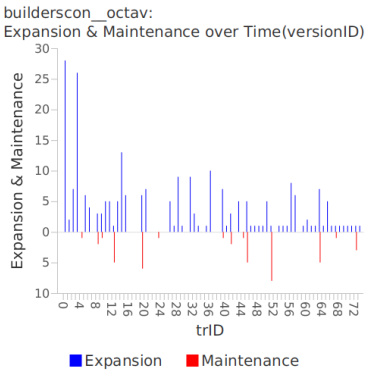
https://github.com/DAINTINESS-Group/Schema_Evolution_Datasets/

Eventually, for each project, we ended up with the automatically extracted
+ time series of changes
+ collected stats on timing, schema size & activity
+ extra statistics manually extracted









Project	#Active commits	#Areeds postV0	#ATurf postV0	Turf absence / presence	DurationIn Days	DurationIn Months	DurationIn Years	#Commits	ActiveCommitRatio	#Tables@Start	#Tables@End	#Attrs@Start	#Attrs@End	TotalTableInsertions	TotalTableDeletions	TotalAttrInsWithTableIns	TotalAttrbDelWithTableDel	TotalAttrInjected	TotalAttrEjected	TatalAttrWithTypeUpd	TotalAttrInPKUpd	TotalExpansion	TotalMaintenance	TotalActivity
nawork__nawork-uri	22	2	20	ABSENCE	3023	100	9	26	85%	1	4	2	23	4	1	30	13	19	15	11	0	49	39	88
jasdel__harvester	7	2	5	PRESENCE	4	1	1	12	58%	4	6	14	17	4	2	14	7	15	19	0	0	29	26	55
curt-labs__GoSurvey	4	1	3	ABSENCE	15	1	1	7	57%	9	11	63	86	2	0	22	0	4	3	8	0	26	11	37
spaceboats__busbus	5	2	3	PRESENCE	17	1	1	8	63%	14	10	107	82	1	5	3	31	11	8	0	4	14	43	57
milogert__ocdns	6	1	5	PRESENCE	38	2	1	9	67%	3	5	10	28	4	2	19	9	10	2	15	0	29	26	55
joyplus__o2oadmin	4	2	2	ABSENCE	43	2	1	10	40%	2	13	18	103	11	0	85	0	0	0	1	0	85	1	86
keybase__node-client	4	1	3	PRESENCE	60	2	1	7	57%	2	2	16	7	3	3	11	21	2	1	1	0	13	23	36
h2oai__steam	8	1	7	PRESENCE	62	3	1	10	80%	24	24	128	128	0	0	0	0	10	10	14	38	10	62	72
CityGrid__twonicorn	7	2	5	PRESENCE	174	6	1	9	78%	17	23	80	131	6	0	46	0	14	9	1	0	60	10	70
hurad__hurad	10	1	9	PRESENCE	287	10	1	15	67%	12	13	105	121	16	15	142	129	4	1	39	0	146	169	315
pw-press__web-project	5	2	3	PRESENCE	293	10	1	11	45%	11	11	72	69	4	4	12	21	18	12	2	0	30	35	65
lamassu__lamassu-scripts	5	1	4	ABSENCE	754	25	3	16	31%	5	7	26	41	2	0	15	0	6	6	0	0	21	6	27
TalkingData__OWL-v3	7	1	6	ABSENCE	792	27	3	14	50%	26	33	171	245	15	8	84	22	51	39	18	0	135	79	214
n2n__rocket	4	1	3	ABSENCE	801	27	3	7	57%	10	10	46	49	5	5	23	21	10	9	10	0	33	40	73
brettkromkamp__topic_db	9	1	8	ABSENCE	821	28	3	16	56%	7	7	37	40	8	8	42	42	12	9	7	4	54	62	116
kronusme__dota2-api	7	2	5	ABSENCE	1008	34	3	15	47%	6	13	74	169	7	0	85	0	13	3	48	0	98	51	149
TwitchScience__rs_ingester	6	1	5	ABSENCE	1126	38	4	8	75%	2	4	9	17	4	2	16	9	4	3	1	0	20	13	33
gugoan__economizzer	6	2	4	ABSENCE	1285	43	4	12	50%	3	8	18	66	16	11	117	74	6	1	1	0	123	76	199
anchorcms__anchor-cms	7	1	6	ABSENCE	1524	51	5	17	41%	10	10	57	59	10	10	57	57	5	3	3	0	62	63	125
GoBelieveIO__im_service	7	2	5	ABSENCE	1586	53	5	9	78%	2	2	5	13	14	14	134	134	8	0	7	0	142	141	283
alextselegidis__easyappointments	8	1	7	ABSENCE	1716	57	5	19	42%	9	11	61	73	2	0	8	0	5	1	21	1	13	23	36
EPICPaaS__appmsgsrv	21	3	18	PRESENCE	63	3	1	24	88%	10	23	81	184	17	4	125	32	14	4	3	0	139	39	178

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

Extracting schema and project histories at the month level

- The ICDE21 data set has been grouped by to provide Monthly Schema Activity (sum of changes performed to the schema, with #changed attributes being the unit of measurement)
- The Monthly Project Activity based on the project histories for each projects, was obtained by:
  - Locally cloning projects from Github;
  - For each commit, the names of the changed files, the date, some extra info on the authors and their msg's;
    - `git log –name-status –nomerges –date=iso`
  - Counted # changed files per commit ;
  - Group by month to produce Monthly Project Activity

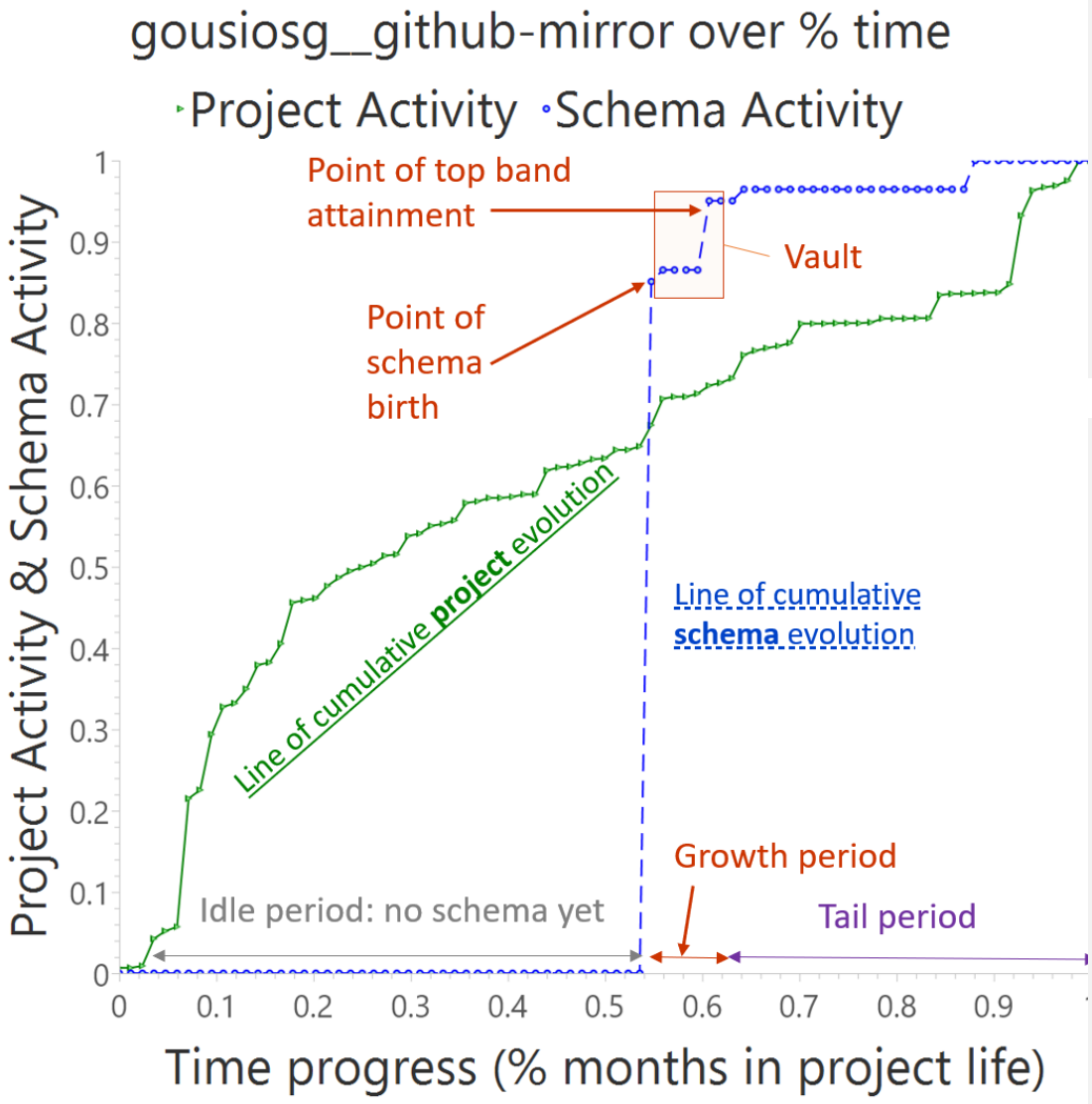
Pattern Extraction Methodology

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/

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) (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.

Quantization of the Measures of Schema Evolution

%Schema Activity at Birth	Low ≤0.25	Middle (0.25 ..0.75]		High (0.75..1)	Full 1
Time of Birth (%PUP)	V0 0	Early (0 .. 0.25]		Middle (0.25..0.75]	Late > 75%
Time of entering top band (%PUP)	V0 0	Early (0 .. 0.25]		Middle (0.25..0.75]	Late > 75%
Interval (%PUP) (birth .. top-band)	Zero 0	Soon (0 .. 0.1]	Fair (0.1 .. 0.35]	Long (0.35 .. 0.75]	Very Long > 75%
Interval (%PUP) (top-band .. end]	Soon ≤0.25	Fair (0.25..0.75]		Long (0.75..1)	Full 1
Active months as %growth	Zero 0	Few (0 .. 0.2]		Fair (0.2 .. 0.75]	High > 75%
Active months as %PUP	Zero 0	Fair (0 .. 0.08]		High (0.08 .. 0.5]	Ultra > 50%

We show (a) how the different metrics have been quantized in labels, and (b) #projects that pertain to each of the produced labels.

- We quantized the different metrics, to be able to perform subsequent analyses
- The quantization was made having in mind the goal to produce general classes of behavior rather than overfit the data set which results in few exceptions in the patterns exactly because the labeling was not done with the patterns in mind.
- The decision of limits was made with the goal of providing labels that (a) are reasonable, and, (b) coarsely divide the metric into labels with similar amounts of projects pertaining to them.
- The extreme values (zero and one) have particular semantics in most cases, too.

Statistical Study

Overall: statistical characteristics of the introduced measures, before studying the corpus breakdown in patterns

Volume of Birth (%Total Change).

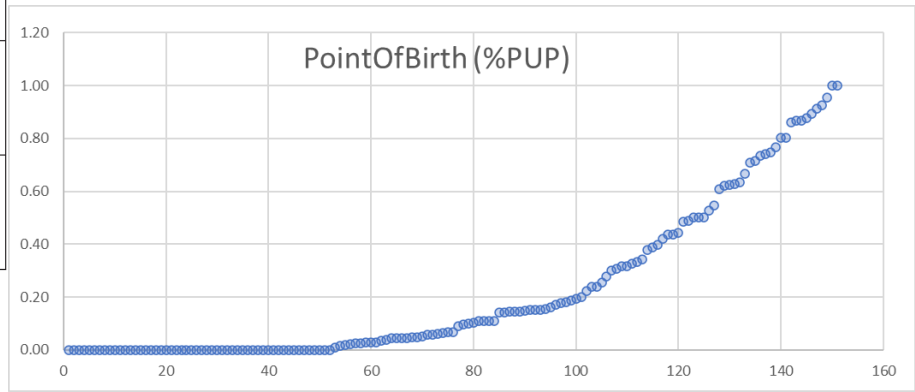
Volume of Birth (%Total Change)	Low <=0.25 (16)	Middle (0.25 ..0.75] (52)		High (0.75..1) (44)	Full 1 (39)
Time Point of Birth (%PUP)	V_p^0 0 (52)	Early (0 .. 0.25] (53)		Middle (0.25..0.75] (33)	Late >75% (13)
Time point of reaching Top Band (%PUP)	V_p^0 0 (23)	Early (0 .. 0.25] (41)		Middle (0.25..0.75] (47)	Late >75% (40)
Interval (%PUP) (birth .. top-band)	Zero 0 (62)	Soon (0 .. 0.1] (26)	Fair (0.1 .. 0.35] (27)	Long (0.35 .. 0.75] (23)	Very Long >75% (13)
Interval (%PUP) (top-band .. end]	Soon <=0.25 (40)	Fair (0 .. 0.25] (48)		Long (0.75..1) (40)	Full 1 (23)
Active months as %growth	Zero 0 (98)	Few (0 .. 0.2] (22)		Fair (0.2 .. 0.75] (22)	High >75% (9)
Active months as %PUP	Zero 0 (98)	Fair (0 .. 0.08] (20)		High (0.08 .. 0.5] (33)	Ultra >50% –

- Reading from right to left, we see that out of 151 projects,
- **39 projects reach Full (100%) activity at schema birth**, and
- 83 projects overall reach High or Full activity at schema birth.
- Overall, more than half of the projects exceed 75% of total activity at schema birth.

Timepoint of Birth (%PUP)

Volume of Birth (%Total Change)	Low <=0.25 (16)	Middle (0.25 ..0.75] (52)		High (0.75..1) (44)	Full 1 (39)
Time Point of Birth (%PUP)	V_P^0 0 (52)	Early (0 .. 0.25] (53)		Middle (0.25..0.75] (33)	Late >75% (13)
Time point of reaching Top Band (%PUP)	V_P^0 0 (23)	Early (0 .. 0.25] (41)		Middle (0.25..0.75] (47)	Late >75% (40)
Interval (%PUP) (birth .. top-band)	Zero 0 (62)	Soon (0 .. 0.1] (26)	Fair (0.1 .. 0.35] (27)	Long (0.35 .. 0.75] (23)	Very Long >75% (13)
Interval (%PUP) (top-band .. end]	Soon <=0.25 (40)	Fair (0 .. 0.25] (48)		Long (0.75..1) (40)	Full 1 (23)
Active months as %growth	Zero 0 (98)	Few (0 .. 0.2] (22)		Fair (0.2 .. 0.75] (22)	High >75% (9)
Active months as %PUP	Zero 0 (98)	Fair (0 .. 0.08] (20)		High (0.08 .. 0.5] (33)	Ultra >50% –

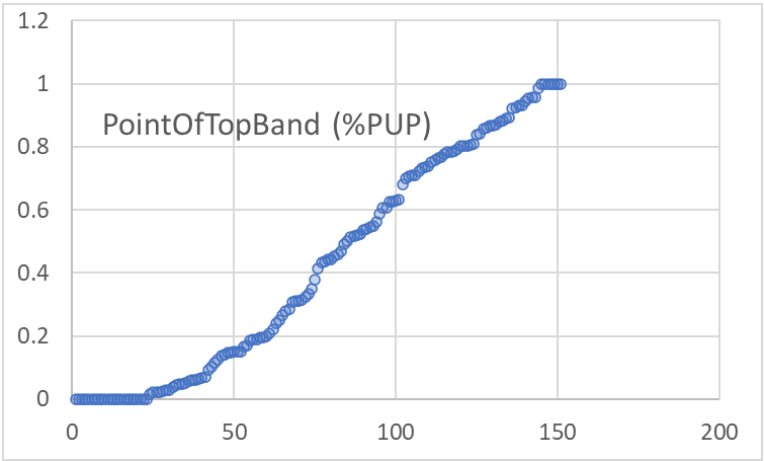
- A large majority of schemata is born early: 52 schemata (one third of the population) are born in V^0.
- Two thirds of the projects (105 projects) see schema birth at V^0 or before 25% of the PUP.
- 74 schemata (half the corpus) are born in the first 10% of time.
- As the distribution of points of birth follows a power-law shape, the rest of the time points form a long tail.



Time point of Entering the Top Band (%PUP)

Volume of Birth (%Total Change)	Low <=0.25 (16)	Middle (0.25 ..0.75] (52)		High (0.75..1) (44)	Full 1 (39)
Time Point of Birth (%PUP)	V_P^0 0 (52)	Early (0 .. 0.25] (53)		Middle (0.25..0.75] (33)	Late >75% (13)
Time point of reaching Top Band (%PUP)	V_P^0 0 (23)	Early (0 .. 0.25] (41)		Middle (0.25..0.75] (47)	Late >75% (40)
Interval (%PUP) (birth .. top-band)	Zero 0 (62)	Soon (0 .. 0.1] (26)	Fair (0.1 .. 0.35] (27)	Long (0.35 .. 0.75] (23)	Very Long >75% (13)
Interval (%PUP) (top-band .. end]	Soon <=0.25 (40)	Fair (0 .. 0.25] (48)		Long (0.75..1) (40)	Full 1 (23)
Active months as %growth	Zero 0 (98)	Few (0 .. 0.2] (22)		Fair (0.2 .. 0.75] (22)	High >75% (9)
Active months as %PUP	Zero 0 (98)	Fair (0 .. 0.08] (20)		High (0.08 .. 0.5] (33)	Ultra >50% –

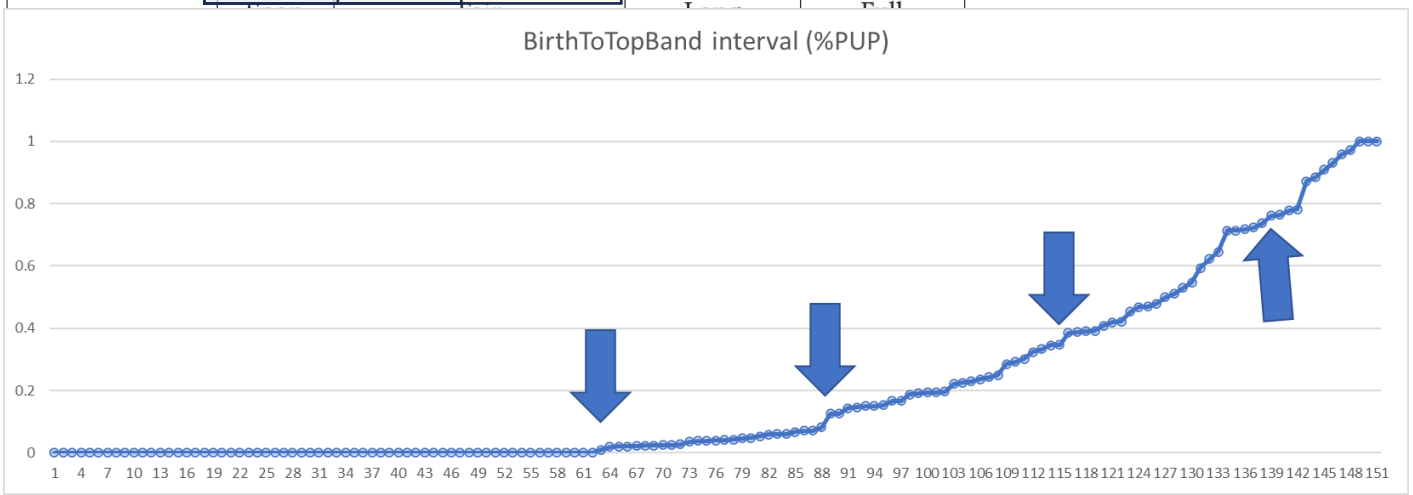
- 64 projects (42%) reached the top band immediately at V^0 (23 of them) or before 25% of the PUP. Another 40 projects (26%) came with a late top-attainment at higher than 75% of the PUP. The rest of the projects were spread in the middle half of the PUP.
- In other words, projects mostly reach top-band soon, although mid-life and late completion of the evolution do exist too.



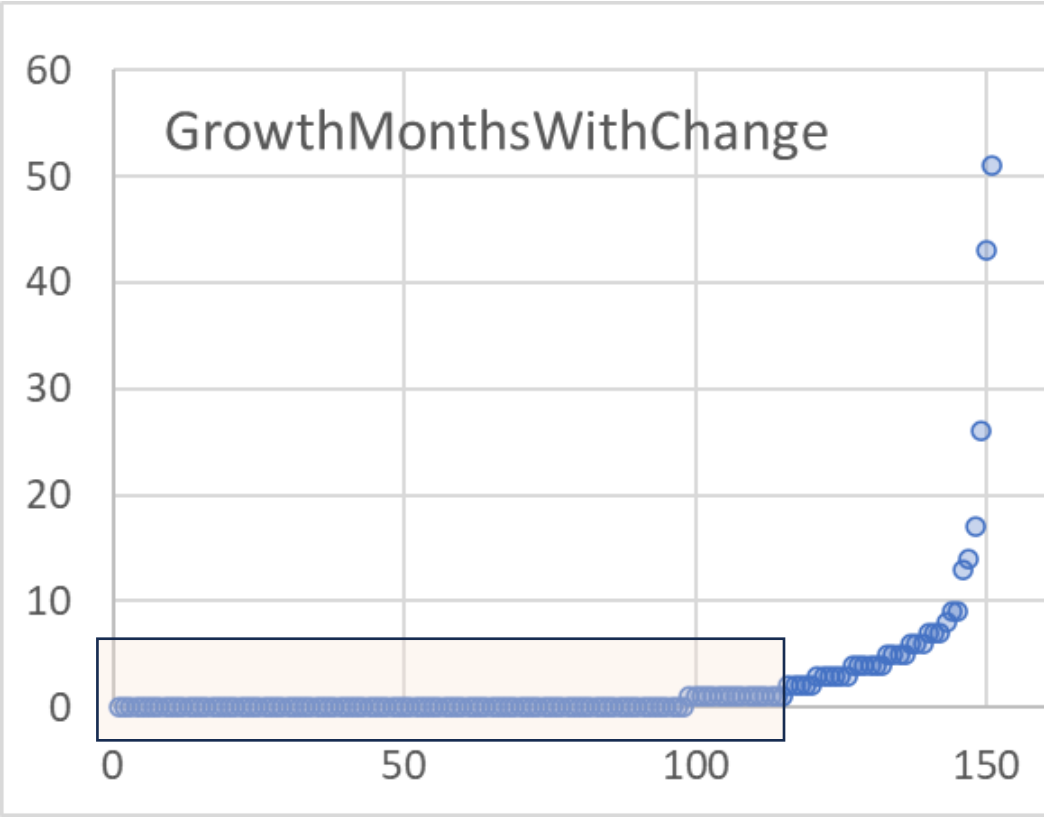
Interval Schema Birth-To-TopBand (%PUP)

Volume of Birth (%Total Change)	Low <=0.25 (16)	Middle (0.25 ..0.75] (52)	High (0.75..1) (44)	Full 1 (39)
Time Point of Birth (%PUP)	V_P^0 0 (52)	Early (0 .. 0.25] (53)	Middle (0.25..0.75] (33)	Late >75% (13)
Time point of reaching Top Band (%PUP)	V_P^0 0 (23)	Early (0 .. 0.25] (41)	Middle (0.25..0.75] (47)	Late >75% (40)
Interval (%PUP) (birth .. top-band)	Zero 0 (62)	Soon (0 .. 0.1] (26)	Fair (0.1 .. 0.35] (27)	Long (0.35 .. 0.75] (23)	Very Long >75% (13)

- **Vaults**. 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.
- 115 (75%) of the projects had an interval of less than 35% PUP.
- The rise from schema birth to the top is therefore mostly fast.

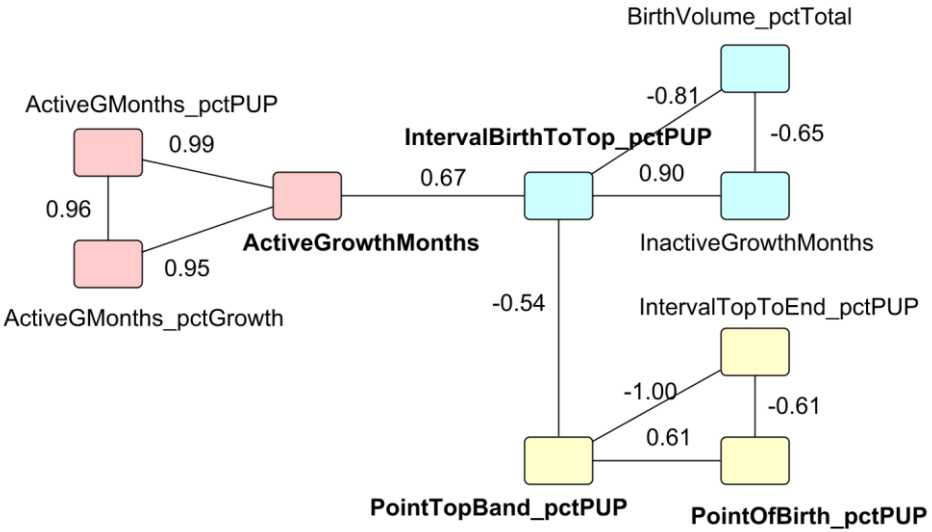


Vaults and Active Growth Months



- **Active Growth Months**. 98 schemata (2/3 of the population) had zero active months in the growth period, from schema birth to the top.
- Another 17 had exactly one month active. Combined, 115 projects, 76% of the population had no more than 1 month of activity from schema birth to top.
- **If there should be a single metric of evidence for the aversion to change, the super-focused nature of schema evolution, and, schema stabilization, Active Growth Months would be the one.**

Correlations



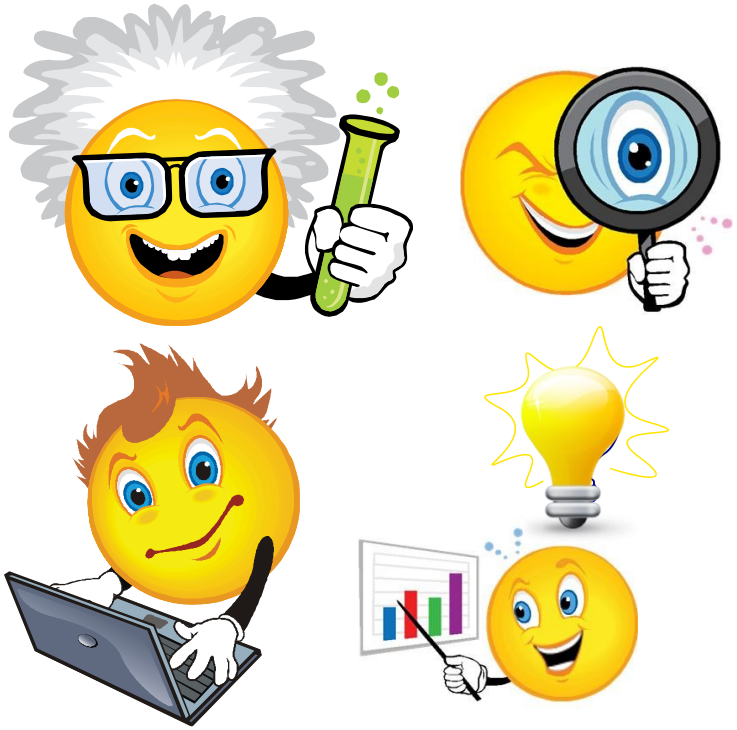
The different measures of Schema Evolution are correlated to some extent.

- The ActiveGrowthMonths (used by default in our analysis) is very tightly related to its normalized versions as percentages of the Project Update Period and the Growth Period.
- Volume of Change at birth and the **Interval from Birth To Top as %PUP are strongly related to the months without change** in the growth period
- Point of Top-Band Attainment and the Interval from the Top-Band to the End of the project are extremely strongly anti-correlated and strongly correlated to Point of Birth (all of this is totally expected)

Aversion to change is predominant: the longer it takes to reach the top-band, the more inactive months you include, i.e., people prefer clustered groups of schema changes rather than constant incremental maintenance.

Summary of simple stats

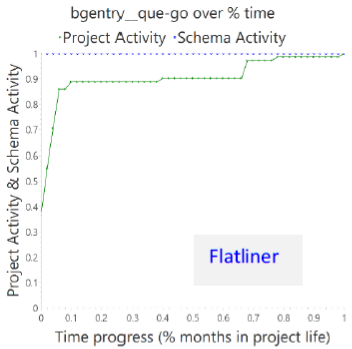
- Birth is mostly done early
  - Two thirds of the projects (105 projects) see schema birth at V0 or before 25% of the PUP.
  - 74 schemata (half the corpus) are born in the first 10% of time.
- Change is mostly sharp in time:
  - 115 projects, 76% of the population had no more than 1 month of activity 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.

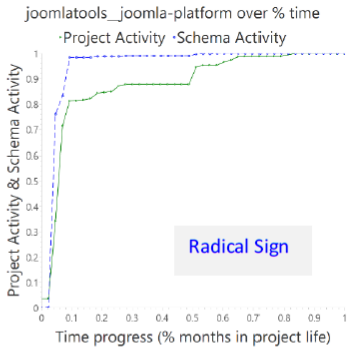


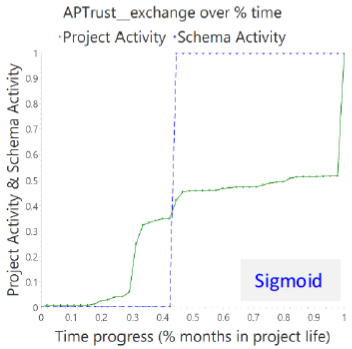
The Patterns

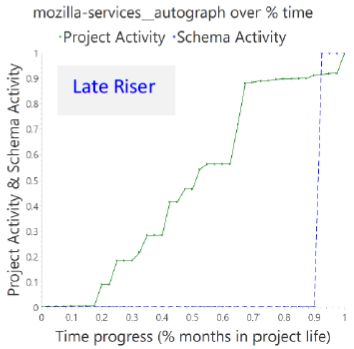
8 Patterns of Change in Time…

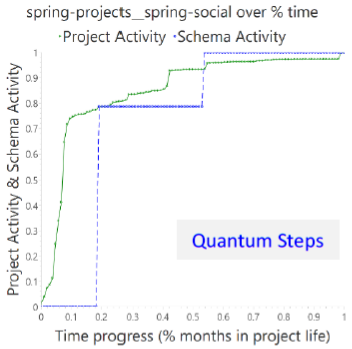
… organized in 3 Families

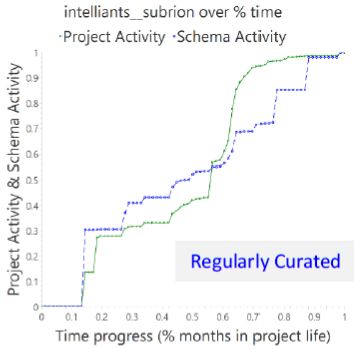


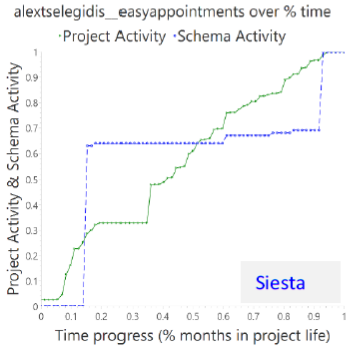


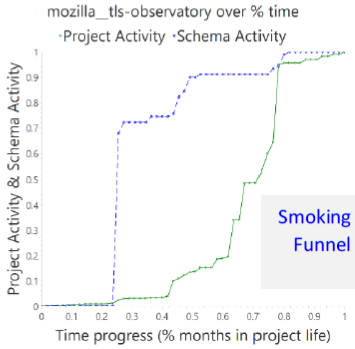




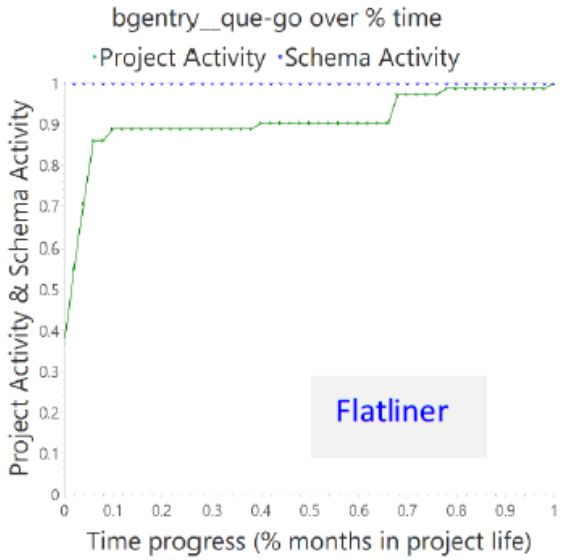


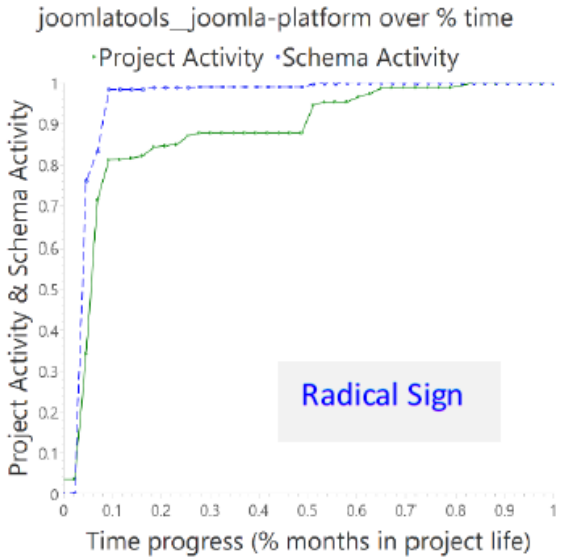


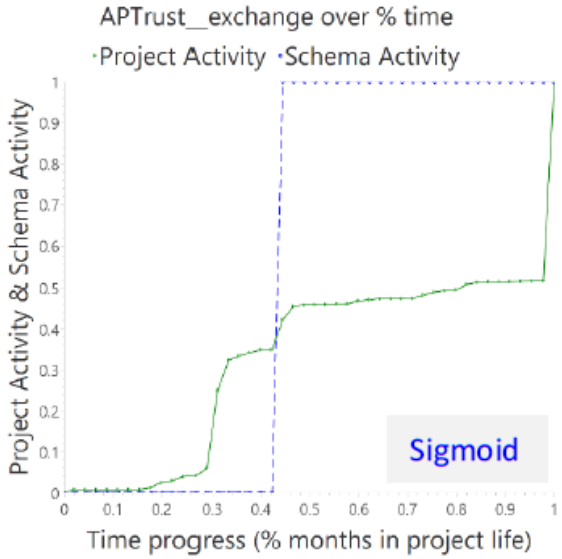


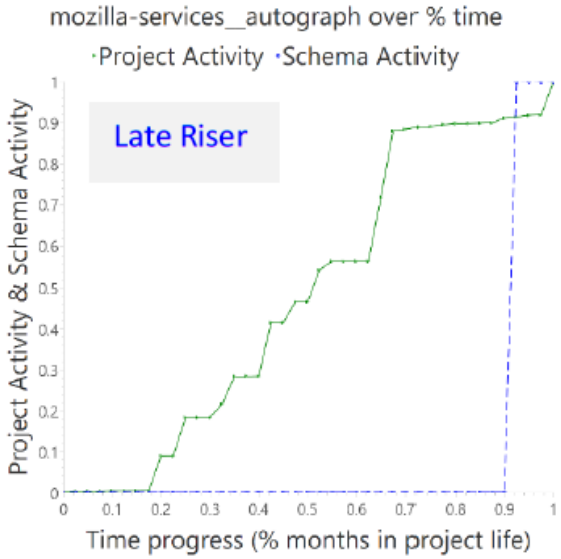


The “**Be Quick or Be Dead**” family of patterns constitutes a family of **very focused change very close to the point of schema birth** - the only difference of the involved patterns is when schema birth takes place.



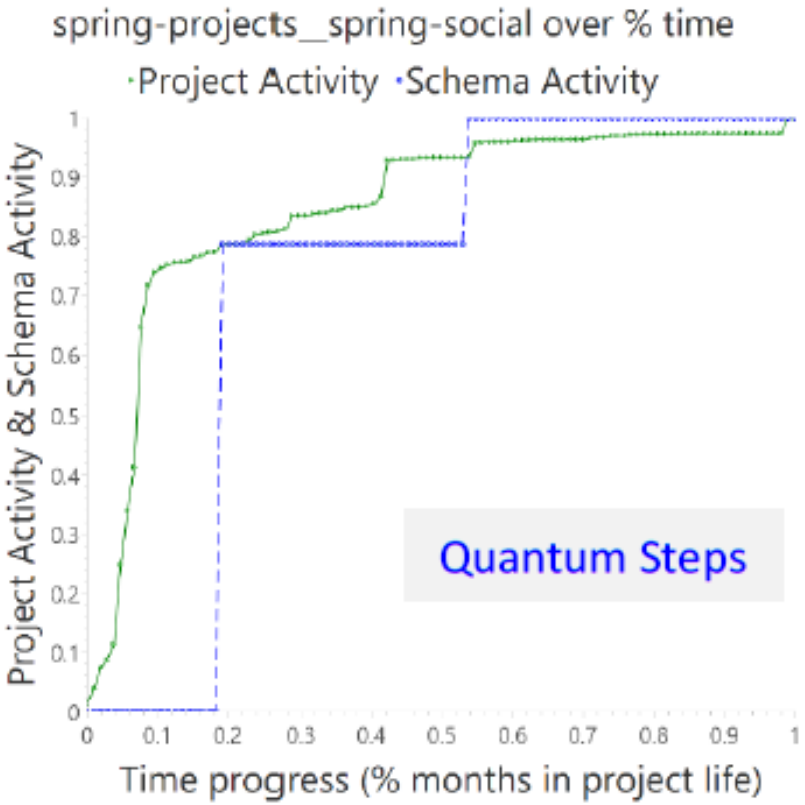


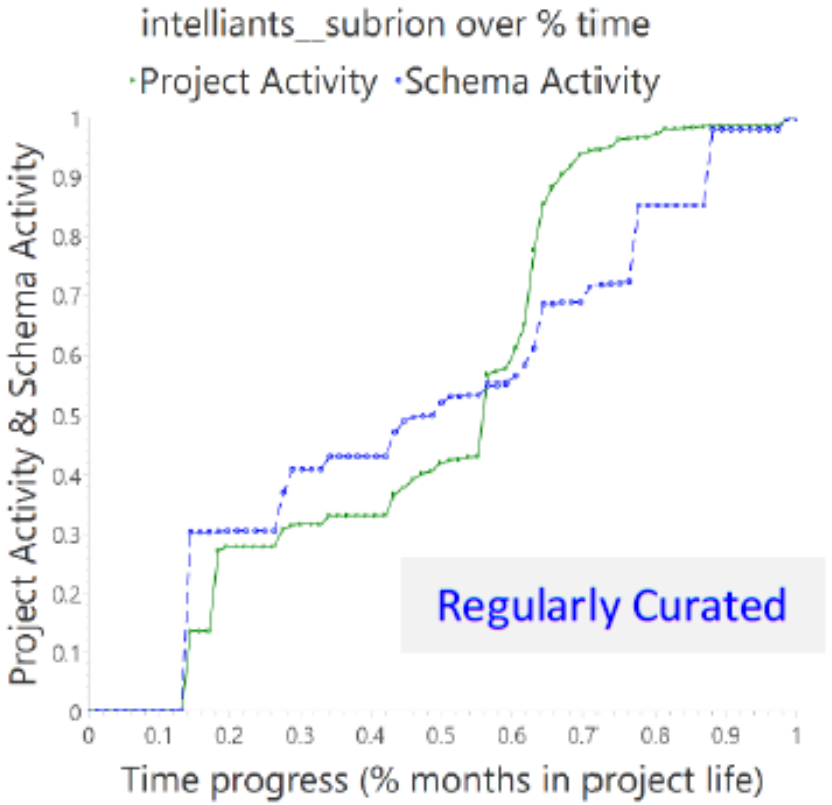




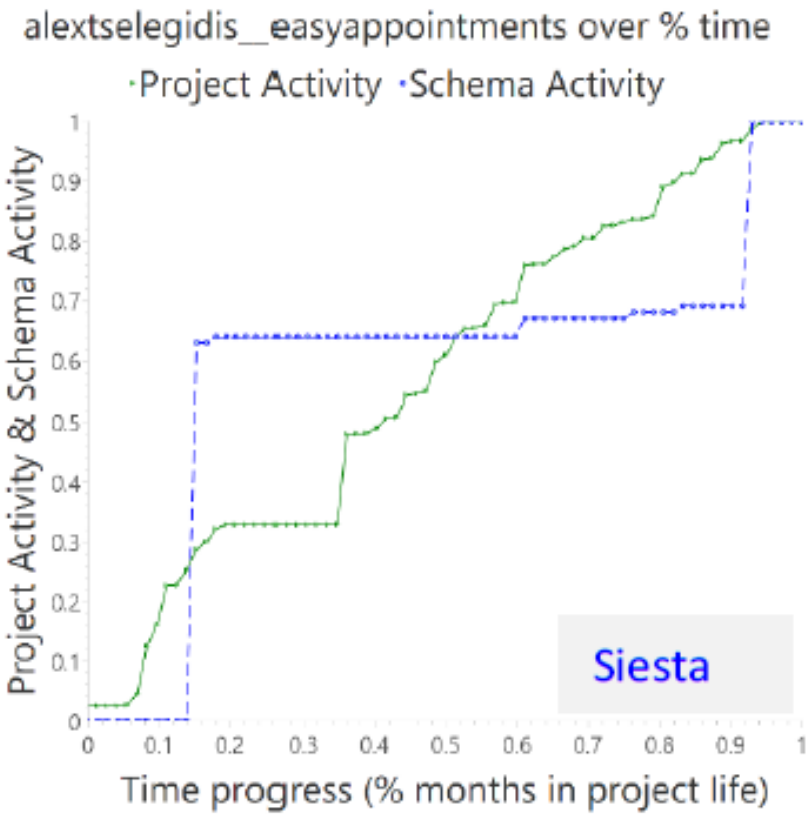
The "**Stairway to Heaven**" family of patterns: both patterns, involve **a fairly regular pattern of change**, with change steps distributed across time.

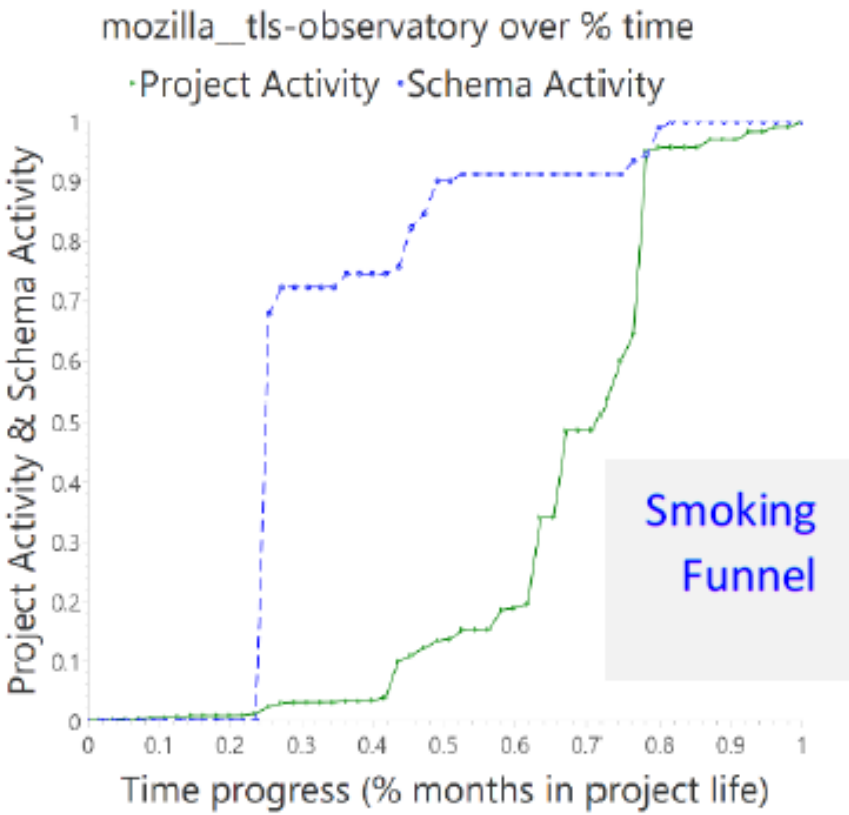
Although **different in the change rate**, both patterns refer to projects that do not reach the top band in a single shot, but, **progressively climb to the top-band over a long period of time**.





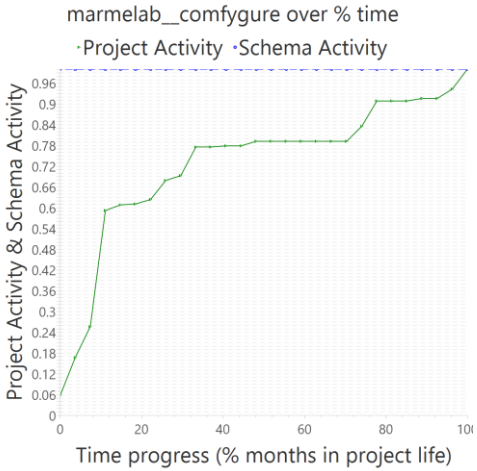
The "**Scared to Fall Asleep Again**" family of patterns: the two patterns, although very different in their characteristics, resemble in that they **include projects where the change is not focused in a single point**, and **happens towards the end of the lifetime of the project**.





Flatliners

Everything happens at birth

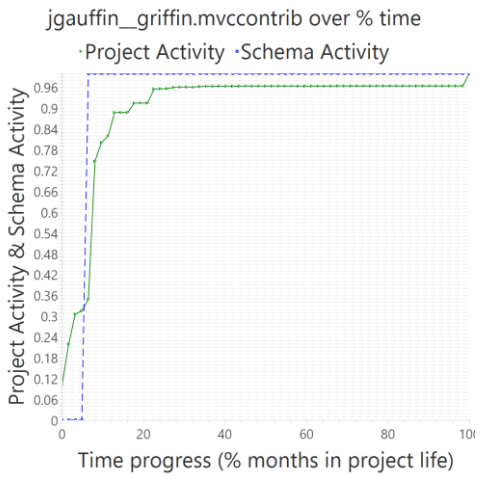


#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% 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

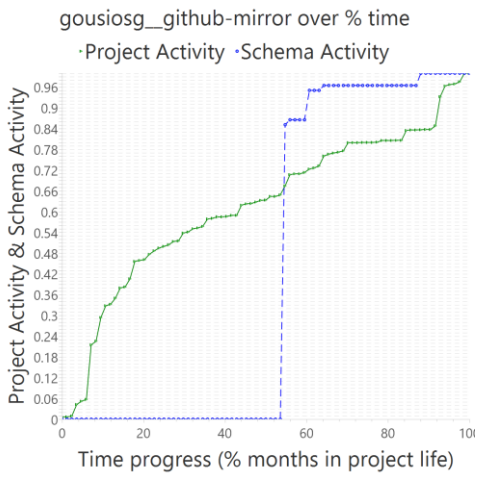
Exceptions: -

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)

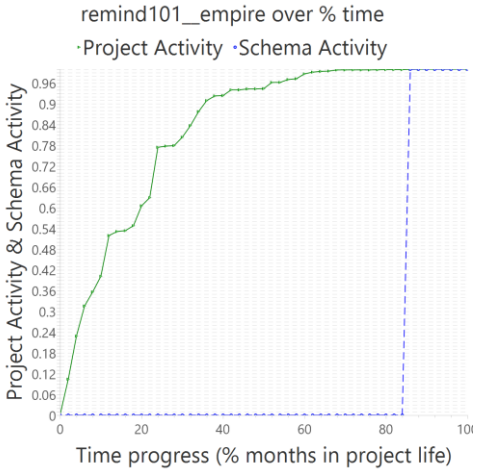


#Prjs	Born	When reaches top band?	How long is growth? (middle life)	How long a tail?	# Active months at growth
Out of 151	*Early <=25%, middle (25% .. 75%], late > 75%*	*Early <=25%, middle (25% .. 75%], late > 75%*	*When /how from birth to top Band (>90% totAct)? 0, soon<=10%, fair<=35%, long <=75% 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)

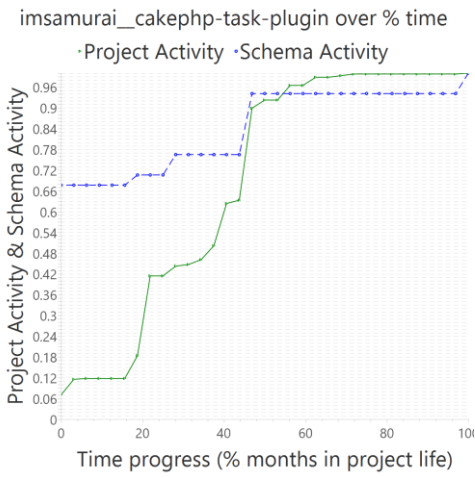
Quantum Steps

A sequence of a few focused steps in the middle…

Two variants, both with few updates:

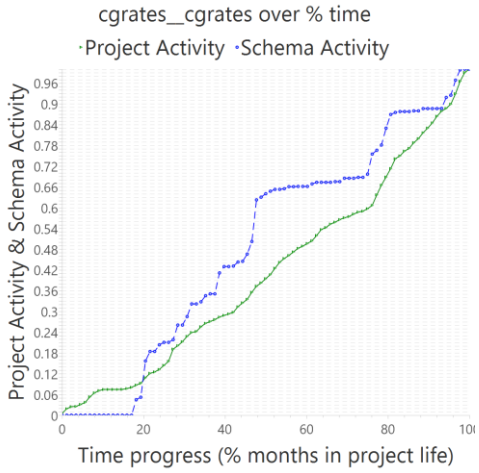
(a) early start, middle top, and,

(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:

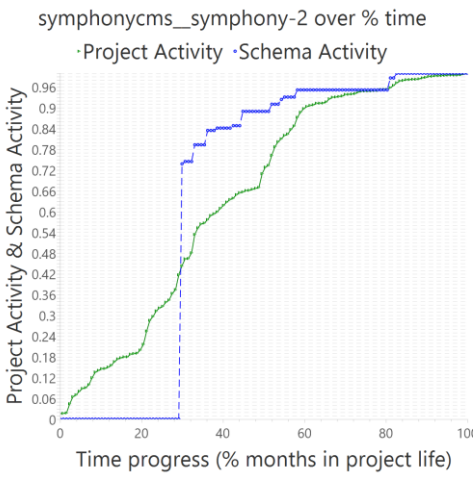
- If born early, reaches top middle or late;
- If born middle, reaches top late

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

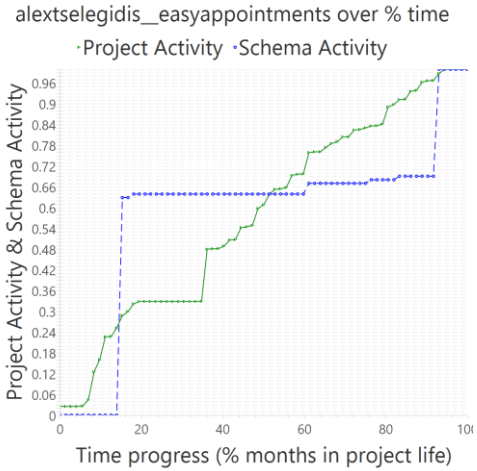
Somewhat late birth, with something like a vault (but not full or super high), and once born, alive with regular schema updates



#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

Pattern Validity

Are these patterns…

VQ1: **genuine** and **reasonable**?

VQ2: (a) **internally cohesive** and (b) **pairwise disjoint**?

VQ3: **generalizable**?

VQ4: **complete**?

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

Validation Questions for this Study

- 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?

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

Is what we see … genuine?

- Genuine character comes **from the process itself**: **all the statistics-related tasks**, like the quantization of the time-related attributes, the pairwise comparison of patterns or the inspection of other attributes **were performed only after the bottom-up manual inspection and classification**
- The **pattern definitions came also after the manual classification was performed**, and had a very small effect to internal restructurings, just for the Regularly Curated family.
  - In line with **Grounded Theory**, a method suited for situations where patterns have to be extracted from collected data via a manual, iterative generation of patterns via comparing every new data item (here: schema history) to all previous patterns, and adjusting the patterns accordingly (see the paper for references).
- The few **exceptions** that exist are also evidence of the genuine human-based, iterative process.

Is what we see … in line with related work?

- “Be quick or be dead" with a single/very few "shot(s)“ & small activity volume:
  - too many projects
  - internal separation: only wrt when the schema was born.
- “Stairway to heaven" with progressive evolution at least until midlife,
  - non-trivial fraction of the projects
  - internal separation: wrt the rate of change time-points
- “Scared to fall asleep again" with signs of late change:
  - a small part of the corpus
  - the two patterns differ at the existence of midlife change hiatus

- Studies of the more distant past with a handful of schemata had to pick schemata that have some change (otherwise there is not much to report), so no aversion to change.
- Studies that work with large numbers of schemata, or studies with necessity sampling of projects, clearly show the aversion to change. This means that
  - both Be quick or be dead && Stairway to heaven are in-line with previous research
  - Scared to Fall asleep again is completely new
- The study of nosql schemata: time behaviors that are visually quite close to the patterns reported here (universality of the patterns?)

Is what we see …reasonable?

- Observed desirable properties in our set of patterns
- First, **few exceptions do exist.** The result of the pattern extraction process was neither perfectly fit to the data, nor with colossal deviations.
- Second, the **patterns**
  - **are disjoint** and
  - **cover significantly (not fully) the space of possible behaviors**.
- Moreover, **each pattern came with a reasonable size**, neither overly dominating the rest, or being insignificant.

	Count of prjs
1_BeQuickOrBeDead	**97**
11_FlatLiner	23
12_RadicalSign	41
13_Sigmoid	19
14_LateRiser	14
2_StairwayToHeaven	**37**
21_QuantumSteps	23
22_RegularlyCurated	14
3_ScaredToFallAsleepAgain	**17**
31_SmokingFunnel	7
32_Siesta	10
Grand Total	**151**

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

Disjointness

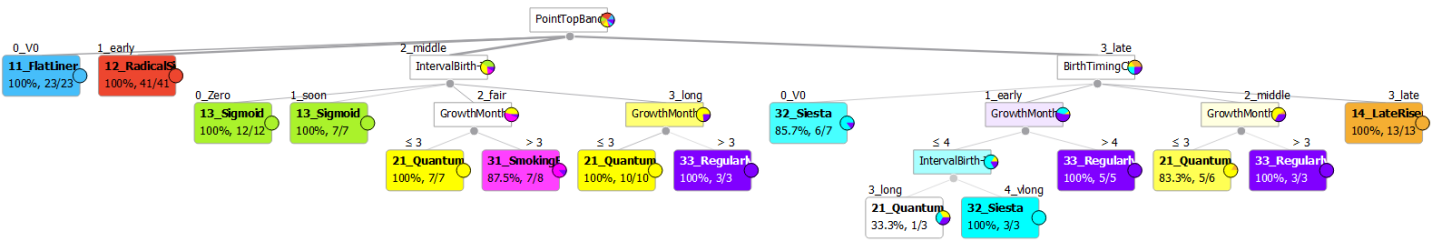
- **Formal Disjointness**: the **formal definitions cover disjoint areas** in the space produced by the Cartesian Product of values for the defining attributes

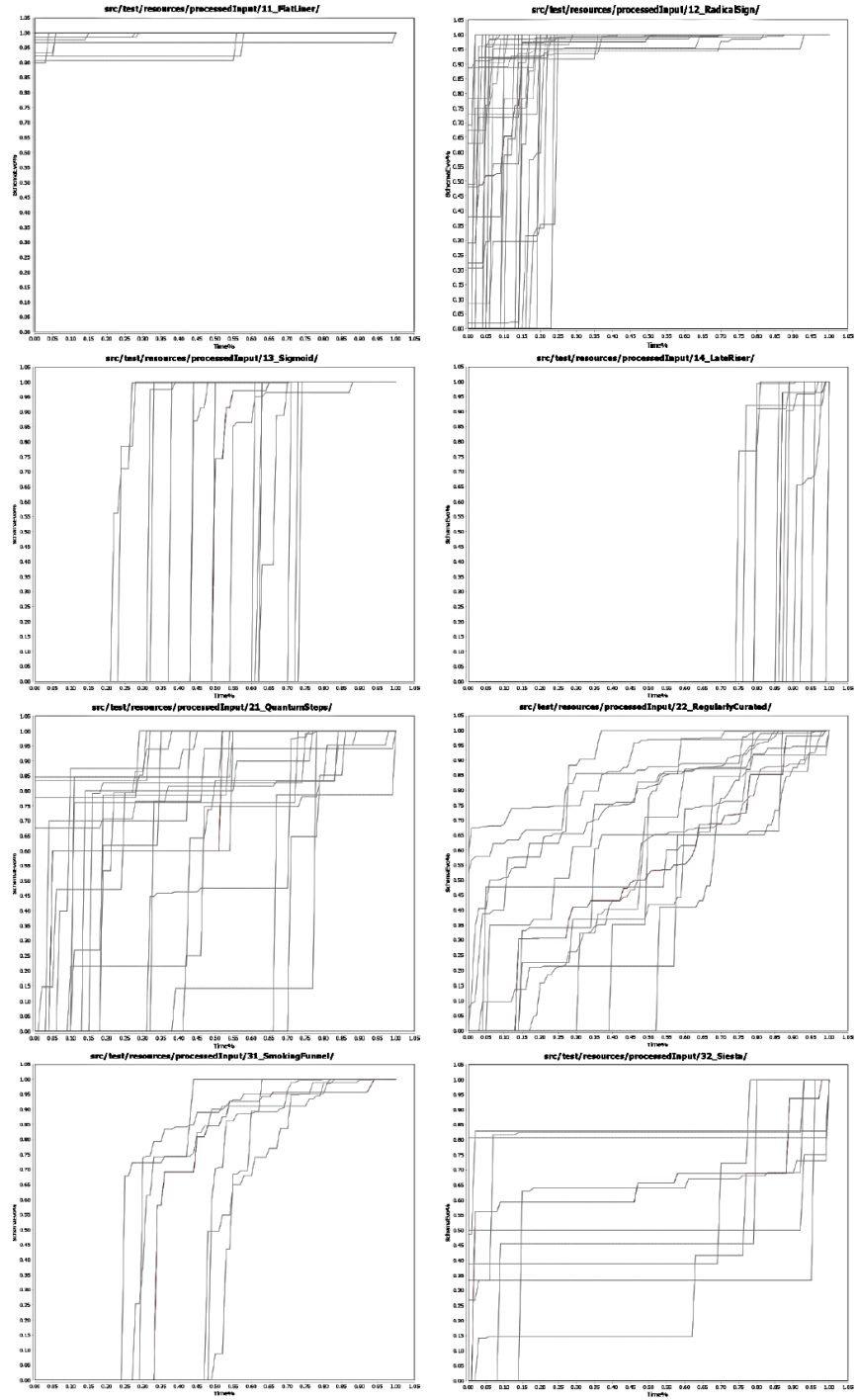
		Birth	*From Growth To the Top Band*							*Tail*
PATTERN FAMILY	**PATTERN**	**Birth Vol Class**	**Birth Timing Class**	**Point Top Band Class**	**Has Single Vault**	**Interval Birth To TopB Class**	**Growth Months With Change**	**Active Pct Growth Class**	**Active Pct PUP Class**	**Interval TopB To End Class**
Be Quick Or Be Dead	**FlatLiner (23)**	high, full	**V_p0**	**V_p0**	TRUE	Zero	0	Zero	zero	full
	Radical Sign (41)	low, fair, high, full	**V_p0, early**	**early**	TRUE, FALSE	Zero soon, fair	0 – 2 *(1 exc)*	any	zero, fair	long *(1 exc)*
	Sigmoid (19)	full, high, fair	**middle** *(2 exc)*	**middle**	TRUE	**Zero, soon**	0 – 1	Zero *(3 exc)*	zero, fair	fair
	Late Riser (14)	high, full *(1 exc)*	**late** *(1 exc)*	**late**	TRUE	**Zero, soon**	0 *(1 exc)*	Zero *(1 exc)*	zero *(1 exc)*	soon
Stairway To Heaven	**Quantum Steps (23)**	high, fair	**V_p0, early** *(1 exc)*	**middle** *(2 exc)*	FALSE	**fair, long**	**0 – 3**	Zero, few *(2 exc)*	zero, fair	fair *(1 exc)*
	(17)	low, fair, high	**middle**	**late**	FALSE	**fair, long**	**0 – 3**	Zero, few, fair	zero, fair, high	soon *(1 exc)*
	(6)									
	Regularly Maintained (14)	low, fair	**V_p0, early** *(3 exc)*	**middle, late**	FALSE	**long vlong**	**>3**	few, fair	high *(1 exc)*	soon, fair
	(11)	fair	**middle**	**late**	FALSE	**fair, long**	**>3**	few, fair	high *(1 exc)*	soon
	(3)									
Scared To Fall Asleep Again	**Siesta (10)**	fair *(2 exc)*	**V_p0, early**	**late**	FALSE	**vlong** *(1 exc)*	**0 – 3** *(2 exc)*	Zero, few	zero, fair, high	soon
	Smoking Funnel (7)	fair *(1 exc)*	**middle**	**middle**	FALSE	**fair**	**>3**	fair, large	fair, high	fair

Disjointness

- **Essential Disjointness**: *are projects "occupying" solely an area of the space of actual values?*
- *To a very large extent yes!* the patterns are focused in a specific area of the domain space and disjoint from the others.
- **Exceptions:**
  - a couple of Siesta projects overlapping with Regularly Curated projects of similar definition,
  - the Quantum Steps and Regularly Curated patterns:
    - although disjoint…
    - … they are the only patterns practically separated by change rate in their growth period …
    - … and span a large area of the domain space of values
- A decision tree is also produced to verify essential Disjointness (with only 4 out of 151 projects that would have been erroneously classified)

PointBirthClass	PointTopBandClass	IntervalBirth-To-TopBand Class	GrowthMonthsWithChange	11_FlatLiner	12_RadicalSign	13_Sigmoid	14_LateRiser	21_QuantumSteps	22_RegularlyCurated	31_SmokingFunnel	32_Siesta	Σ
0_V0	0_V0	0_Zero	0-3	23								23
	1_early	1_soon	0-3		8							8
		2_fair	0-3		7							7
			> 3		1							1
	2_middle	2_fair	0-3					2				2
		3_long	0-3					2				2
			> 3						2			2
	3_late	4_vlong	0-3								6	6
			> 3						1			1
1_earl	1_early	0_Zero	0-3		15							15
		1_soon	0-3		9							9
		2_fair	0-3		1							1
	2_middle	1_soon	0-3			2						2
		2_fair	0-3					5				5
			> 3						1			1
		3_long	0-3					7				7
			> 3						1			1
	3_late	3_long	0-3					1			1	2
			> 3						3			3
		4_vlong	0-3								1	1
			> 3						3		2	5
2_mid	2_middle	0_Zero	0-3			12						12
		1_soon	0-3			5						5
		2_fair	> 3							7		7
		3_long	0-3					1				1
	3_late	1_soon	0-3				1					1
		2_fair	0-3					2				2
			> 3						1			1
		3_long	0-3					3				3
			> 3						2			2
3_late	3_late	0_Zero	0-3				12					12
		1_soon	> 3				1					1
Σ				23	41	19	14	23	14	7	10	151





- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

Visual inspection

Pattern cohesion refers to the internal homogeneity of the projects that pertain to each pattern.

Our first attempt towards producing cohesive patterns was via visual inspection.

The visual inspection of the schema evolution progress of the different patterns is quite revealing on the similarity between the members of each pattern.

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

Cohesion: Just a few exceptions exist

- Two projects classified as Sigmoid violate the "middle-born" part of the definition by being born early.
- A Late Riser project reaches the top band in middle life, violating the requirement of late attainment of top-band.
- Siesta has 2 projects exceeding the 0-3 months growth activity in the end, and a project that reaches growth just ’long’ after schema birth (and not ’very long’).
- A Quantum Step project reaches top late rather than middle.

Pattern	#prjs	Exceptions	Overlaps
Flatliner	23	–	–
Radical Sign	41	–	–
Sigmoid	19	2	–
Late Riser	14	1	–
Quantum Steps	23	2	–
Regularly Curated	14	–	–
Smoking Funnel	7	–	–
Siesta	10	3	–

See report in the accompanying web site of the paper

Cohesion: quantitative metrics

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

- We quantized each project's time series to a vector of 20 measurements, one for each interval of 5% of time (i.e., at 0%, 5%, 10%, . . . of time), and computed the centroid for the corpus of each pattern.
- We assessed the SSE and the Mean Distance to Centroid per pattern

Pattern	#prjs	SSE	MDC
11_FlatLiner	23	0.18	0.06
12_RadicalSign	41	23.25	0.70
13_Sigmoid	19	31.76	1.26
14_LateRiser	14	9.33	0.78
21_QuantumSteps	23	36.20	1.16
22_RegularlyCurated	14	12.64	0.90
31_SmokingFunnel	7	5.00	0.83
32_Siesta	10	9.15	0.91

$$SSE(C_i) = \sum_{x \in C_i} dist(x, c_i)^2$$

$$MDC(C_i) = \frac{1}{m_i} \sum_{x \in C_i} dist(x, c_i)$$

- C_i refers to cluster *i*, with a center c_i and m_i data points
- *x* is a data point assigned to C_i
- *dist(a,b)* is a distance function between two data points, which, unless stated otherwise, we assume to be the Euclidean distance

Threats to validity: External Validity

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

- External validity: we argue that the larger dataset is a very good representative of significant FOSS projects
  - GitHub is the main public repository for FoSS prj's.
  - We applied the filter of more than one contributor, more than 0 stars and non-forking
  - Subsequently, filtered out tests, examples & demos
  - Project domains include Content Management Systems, IoT Management on the cloud, Task Management Systems for O/S's, Messaging Platforms, Systems for the management of Scientific Data, Web on-line stores, On-line Charging Systems (OCS)…
- Limitations:
  - Multi-vendor DDL: covered only one vendor
  - Non-sql schemata, non .sql suffixes, multi-file DDL, incremental definitions of DDL

Threats to validity: External Validity

- **External validity**:
  - The **corpus** of 151 significant projects is **substantial** and (b) suitable: with more than 12 months duration the projects hade ample time for the patterns to appear.
  - The patterns produced are **cohesive**, **disjoint** and **not overfitted** to the data set used.
  - This means **we can generalize our results to the broader scope of schema evolution for FOSS**

Threats to validity: Experimental Validity

- We tested our extraction scripts with OpenCart, the largest of our studied projects for which we had a previous past extraction of its history, in 2016.
  - almost identical result, as only one commit out of 412 was missing from the GitHub history we extracted.
- 100% match for manual test of the histories of the retrieved files for a random sample of 50 cases.
- 100% match for removed projects from GitHub at the time of the cloning via a sample of 7 of them.
- Concerning our own software, we did extensive checks to our metrics computation tools.

Threats to validity: Experimental Validity

- Experimental validity:
  - We have checked our change detection and chart-generating software via tests and code reviews & trust it.
  - We have iteratively rechecked our manual allocation of projects to patterns

- Genuine?
- Cohesive & disjoint?
- Generalizable?
- Complete?

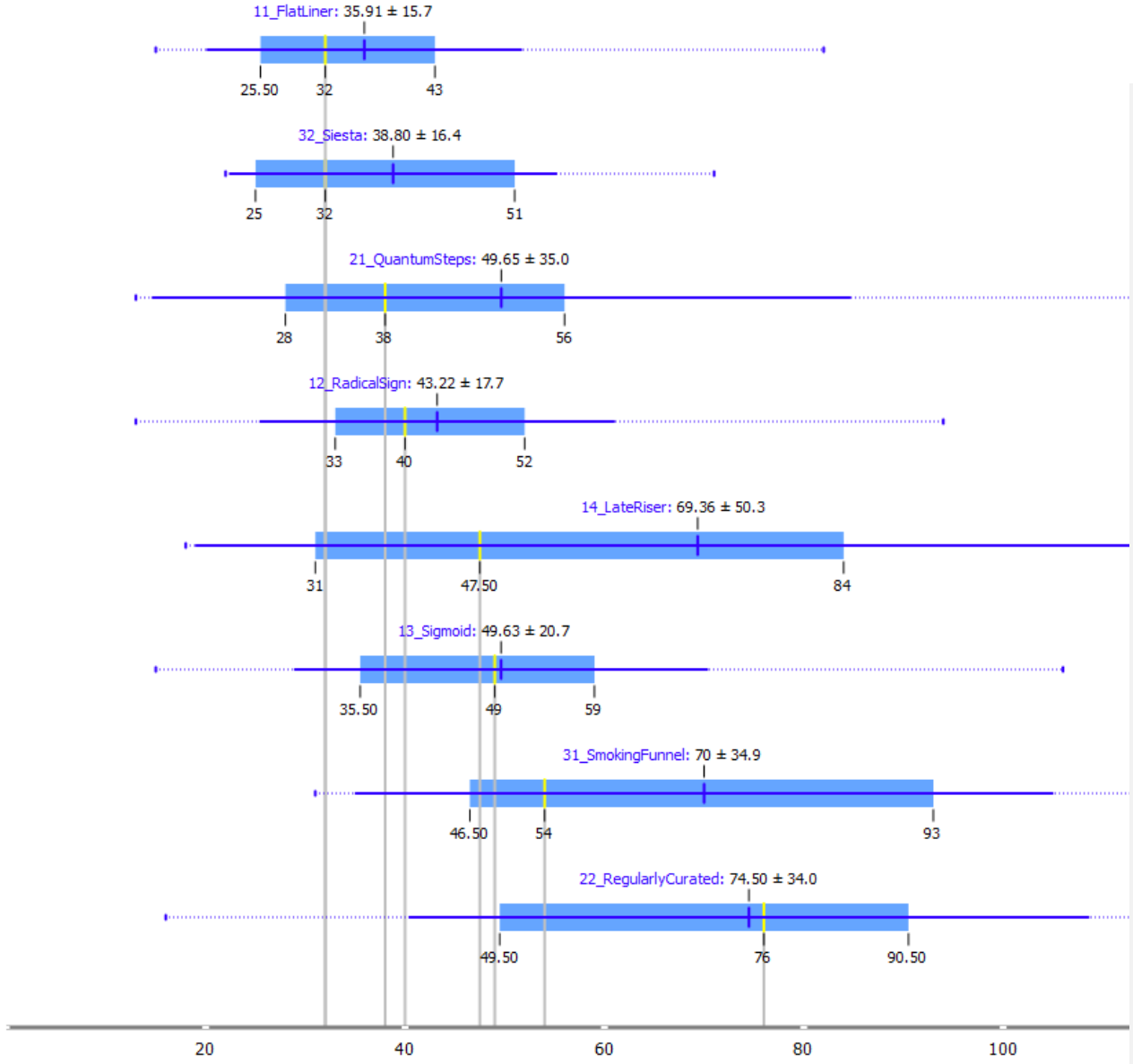
Is what se see … complete?

- We cannot exclude the possibility that other patterns of change do exist
- Still, the overall situation hints that the existence of other patterns is rather unexpected:
- A large part of the space of possible value combinations is already covered.
- Second, although there do exist value combinations several of them are unattainable
  - e.g., a project with late schema birth, is obligatorily restricted to have a late top-band attainment and a short tail.

Patterns and their characteristics

Explicit relationship of patterns to schema evolution quantitative characteristics

Project Update Period (PUP) in months

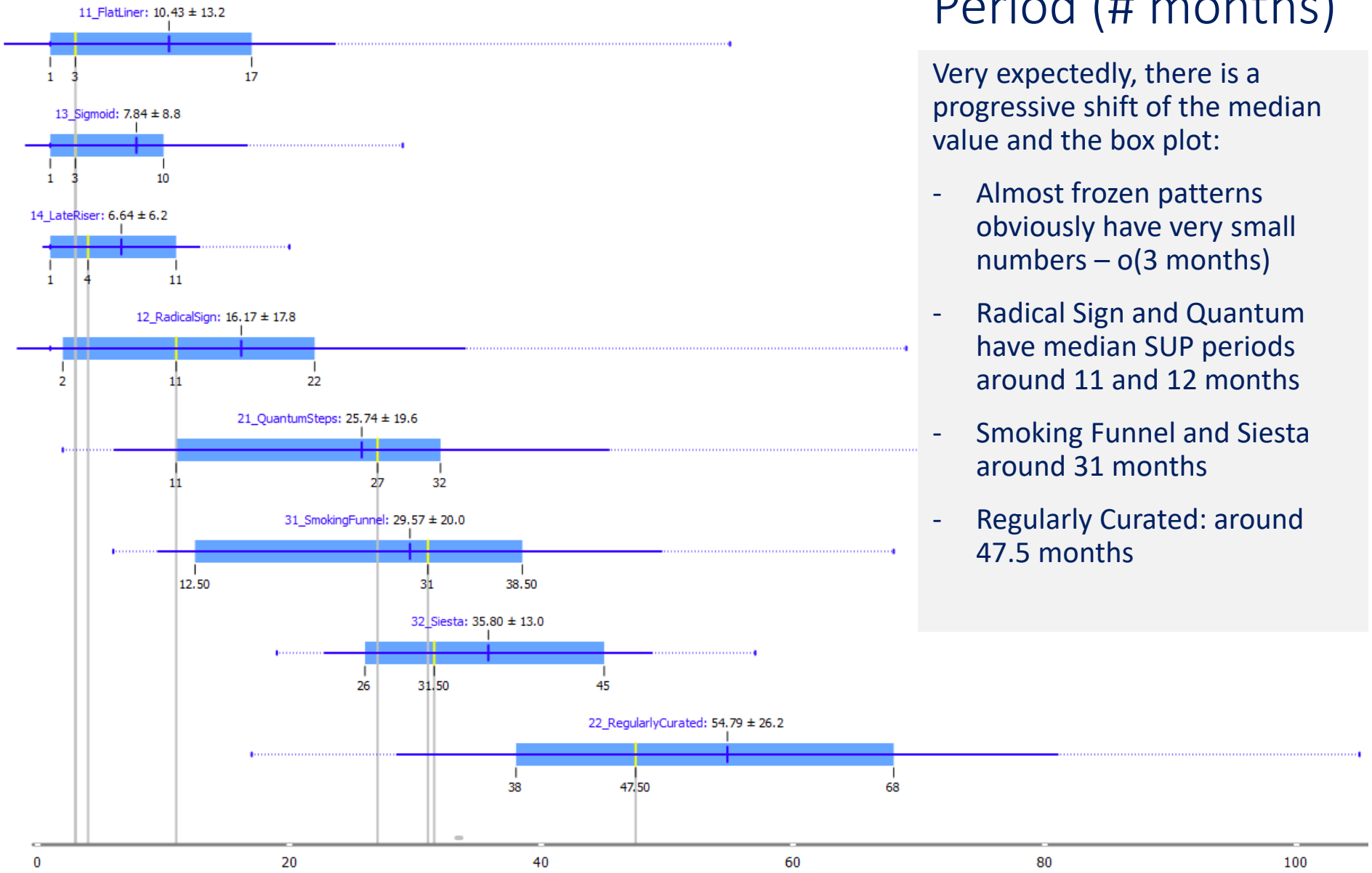


It is extremely revealing that **almost all patterns have median values of the Project Update Period that are very close**.

This means that **the behavior of the schema evolution does NOT relate to the duration of the project** – i.e., there is no "linear" (or other) relationship between how long the project lasted, wrt how its schema evolved!

The only half-exception seems to be Regularly Curated projects, but only wrt the median – the boxplot is very much included within the box plots of Late Risers and Smoking Funnels

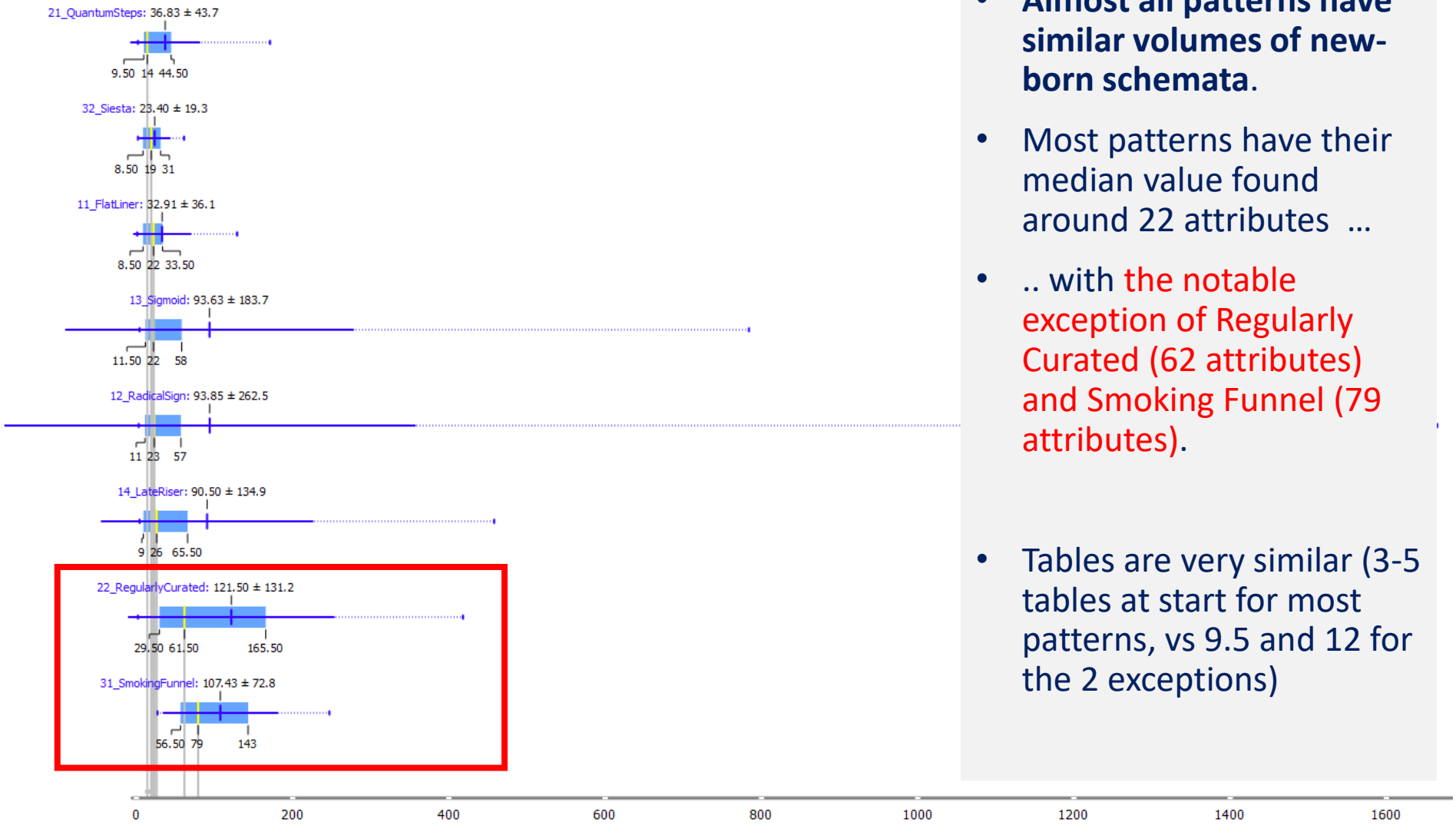
Schema Update Period (# months)



Very expectedly, there is a progressive shift of the median value and the box plot:

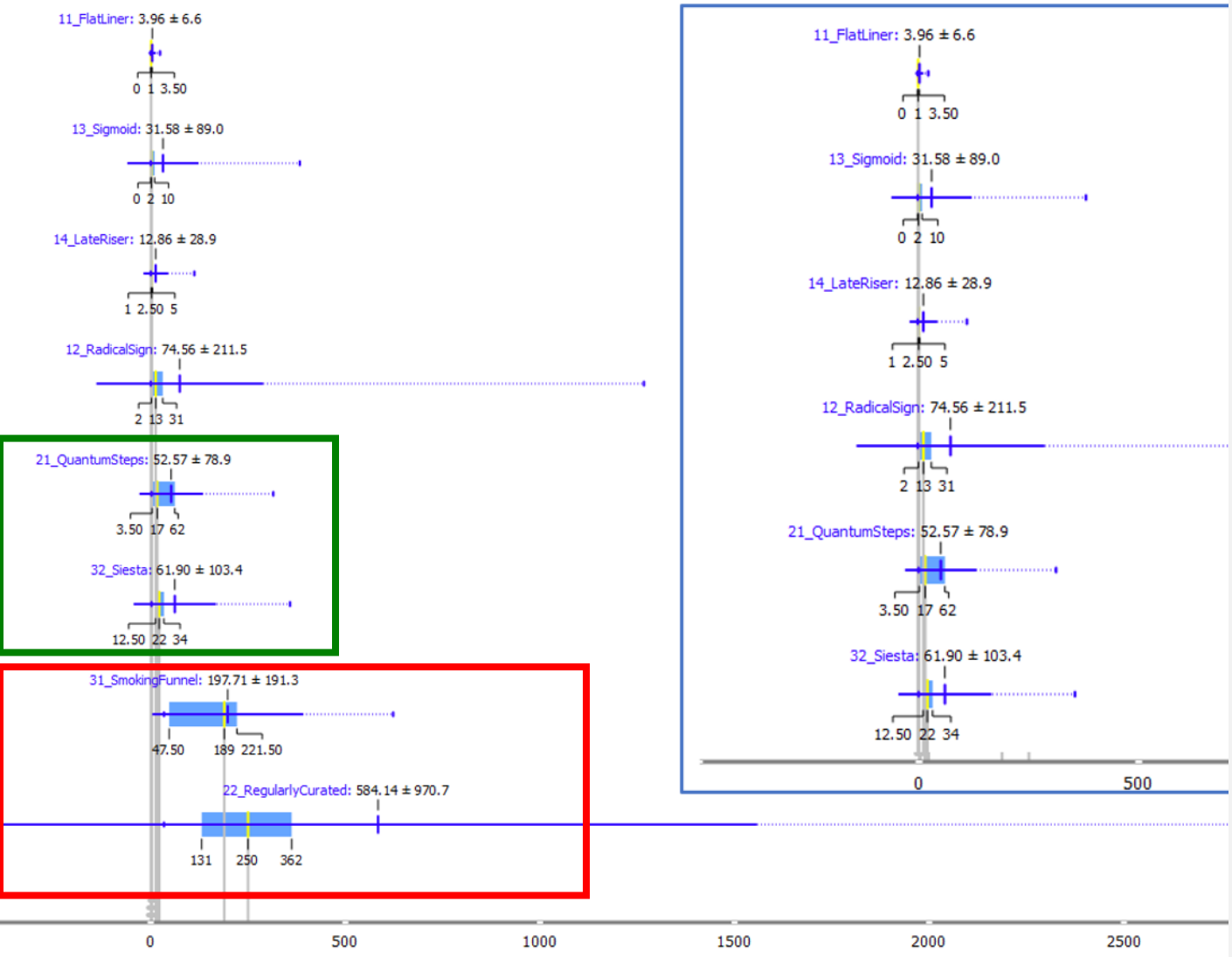
- Almost frozen patterns obviously have very small numbers – o(3 months)
- Radical Sign and Quantum have median SUP periods around 11 and 12 months
- Smoking Funnel and Siesta around 31 months
- Regularly Curated: around 47.5 months

Schema Size at Birth (attributes)



- **Almost all patterns have similar volumes of new-born schemata**.
- Most patterns have their median value found around 22 attributes …
- .. with the notable exception of Regularly Curated (62 attributes) and Smoking Funnel (79 attributes).
- Tables are very similar (3-5 tables at start for most patterns, vs 9.5 and 12 for the 2 exceptions)

Total Activity



Total Schema Activity (as well as **Total Schema Expansion** and **Total Schema Maintenance** which typically follow the pattern of Total Activity) **is rather indifferent to the patterns.**

Change values are small …

.. except for Smoking Funnel and Regular Curation (the two smallest patterns in terms of population), who have larger values.

Progressive shift from

a) top four, “almost frozen” patterns,

b) Siesta and Quantum steps with somewhat higher medians, and,

c) Smoking Funnel and Regular Curation which come with orders-of-magnitude higher values.

<u>However, this monotonicity does not change the fact that the overlap of the fairly narrow IQR’s, at low values is significant.</u>

Total Activity

- **Why do we see this behavior?** It is important to highlight that the behavior towards schema evolution is not obligatorily in sync with the behavior towards source code evolution. For example, even in really frozen patterns, like, e.g., the populous Radical Sign, we can see vast ranges of PUP values. Similarly, for the schema size at birth.
- **Why Smoking Funnel and Regularly Curated projects are different?** These two include the most active projects in terms of total amount of change.

Summary of how patterns differ wrt evolution characteristics

- Almost all patterns have median values of the Project Update Period that are very close, i.e., their evolution does NOT relate to the duration of the project
- Almost all patterns have similar volumes of new-born schemata, except for Regularly Curated and Smoking Funnel that have significantly larger mean values of birth volume
- Total Schema Activity (as well as Total Schema Expansion and Total Schema Maintenance which typically follow it) is rather indifferent to the patterns, with small change values except for Regularly Curated and Smoking Funnel who have larger values than the rest.
- Thus, Smoking Funnel and Regularly Curated projects start bigger and contain higher amounts of schema evolution activity than the rest of the projects who start small and typically show lower values of change.

Predicting Patterns

Can we predict which pattern a project might follow, given some early characteristics?

Predicting schema evolution is an extremely difficult problem

- To the best of our knowledge, there is no attempt to the problem so far for two reasons.
  - To a very large extent, the evolution depends heavily on the particularities of the development and curation team, along with the idiosyncrasies of the project itself. Thus, predictions would require a very detailed charting of both project and anthropocentric characteristics.
  - A really large corpus of schema histories is needed to come up with some broad approximation from a statistical point of view.
- We make such a preliminary attempt in this paper, to give a broad overview of how the point of schema birth and the future behavior of a project in terms of its pattern of schema evolution re-late.
- Several attempts towards further refinements (e.g., in the form of decision trees) have not provided better insights (OK, “not yet” ☺).

Predictions are always hard, esp., for the future

Research Question: "Assume a curator, or an external assessor, who extracts (e.g., via git log and out tool set) the history of changes of a software project and its relational database. Can the curator make an educated guess on the future of how the schema will evolve?"

	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%

Predicting schema evolution

Assuming that the schema is born during M0 of the project life, the probability that the schema will be

- completely frozen is 75% (!), as a flatliner or a radical sign.
- under steady regular curation via the Quantum Steps or Regular Curation patterns, is 14%.
- 1/3 of the corpus projects have been born at month zero.

	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%

Predicting schema evolution

- Assuming the schema is born in the next six months, there is …
- … a 53% chance that the schema will follow a sharp, focused evolution, and,
- … a 40% chance it will follow a regularly curated life.
- 90 projects out of 151, i.e., 60% of the corpus, have already been born in the first six months of the projects' lives.

	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%

Predicting schema evolution

- Assuming the schema is born in months 7 - 12, there is …
- a drop in the number of projects born (13) …
- … but with similar distribution of probabilities for the families.
- The only significant change is that the probability of a more active regular curation rises to 23%.
- 2/3 of the corpus' projects have the schema being born within the first year.

	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%

- As the point of schema birth exceeds the first year, there is
- … a 64% chance of a sharp focused change,
- … a 21% chance for a regular maintenance, and
- … a 15% chance for a smoking funnel behavior.

Predicting schema evolution

	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%

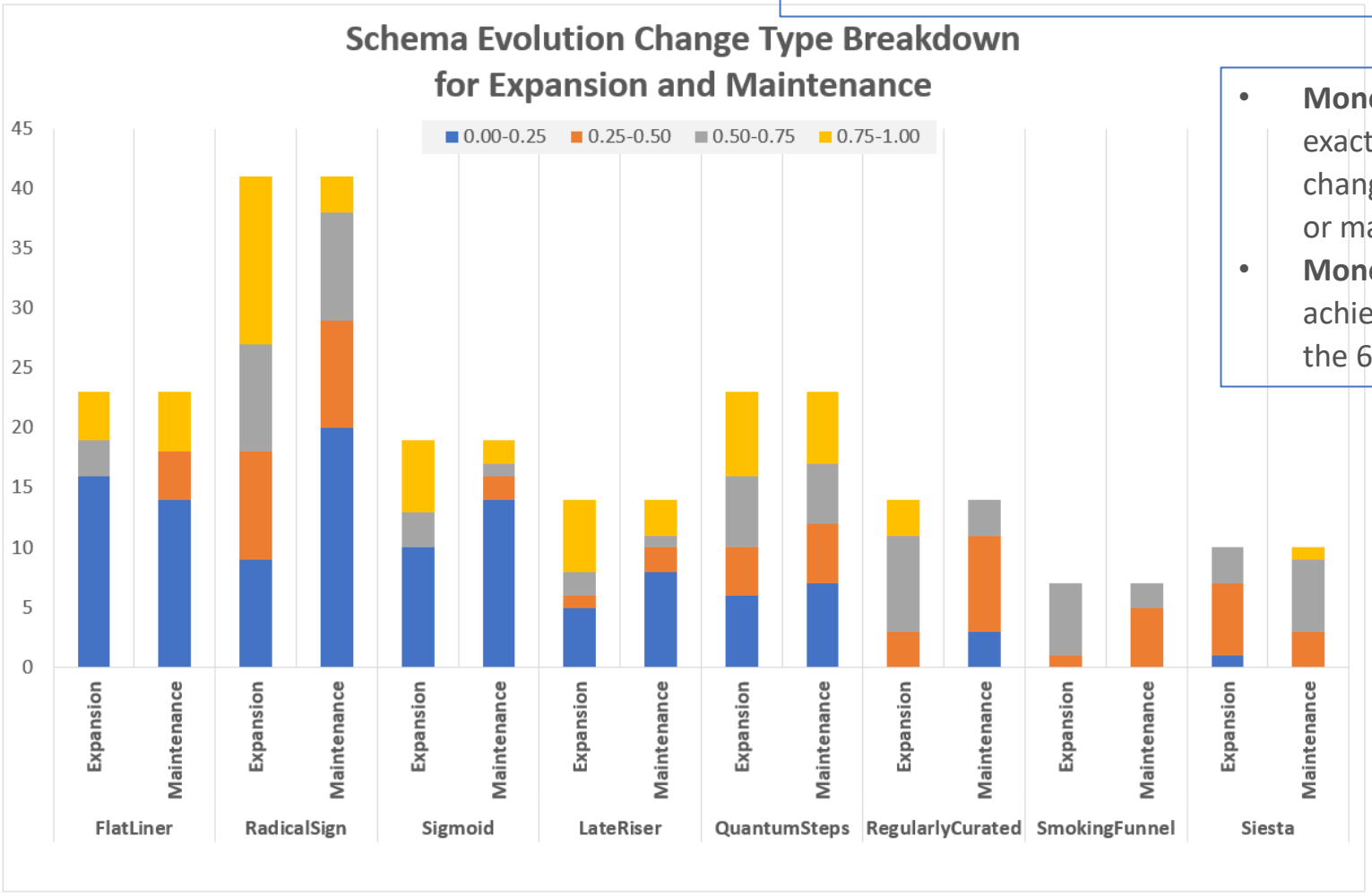
Mixture of change types

What does schema evolution consist of?

See report in the accompanying web site of the paper

Types of Change (#attributes)

Expansion	• attributes born with a new table
	• attributes injected into an existing table
Maintenance	• attributes deleted with a removed table
	• attributes ejected from a surviving table
	• attributes having a changed data type
	• attributes involved with a changed primary key
Total Activity	Expansion + Maintenance



- **Monothematic change**: exactly one mega-type of change, i.e., only expansion or maintenance.
- **Monotype change** is change achieved via exactly one of the 6 types of changes.

	Expansion	Maintenance	#prj	%prjs of pattern
11_FlatLiner	0%	0%	11	48%
23	0%	100%	5	22%
	(0%-40%)	(60%-100%)	0	0%
	middle	level	1	4%
	(60%-100%)	(0%-40%)	4	17%
	100%	0%	2	9%
12_RadicalSign	0%	0%	6	15%
41	0%	100%	1	2%
	(0%-40%)	(60%-100%)	6	15%
	middle	level	9	22%
	(60%-100%)	(0%-40%)	9	22%
	100%	0%	10	24%
13_Sigmoid	0%	0%	8	42%
19	0%	100%	1	5%
	(0%-40%)	(60%-100%)	1	5%
	middle	level	2	11%
	(60%-100%)	(0%-40%)	3	16%
	100%	0%	4	21%
14_LateRiser	0%	0%	2	14%
14	0%	100%	3	21%
	(0%-40%)	(60%-100%)	0	0%
	middle	level	2	14%
	(60%-100%)	(0%-40%)	1	7%
	100%	0%	6	43%
21_QuantumSteps	0%	0%	0	0%
23	0%	100%	4	17%
	(0%-40%)	(60%-100%)	4	17%
	middle	level	3	13%
	(60%-100%)	(0%-40%)	8	35%
	100%	0%	4	17%
22_RegularlyCurated	0%	0%	0	0%
14	0%	100%	0	0%
	(0%-40%)	(60%-100%)	0	0%
	middle	level	7	50%
	(60%-100%)	(0%-40%)	7	50%
	100%	0%	0	0%
31_SmokingFunnel	0%	0%	0	0%
7	0%	100%	0	0%
	(0%-40%)	(60%-100%)	0	0%
	middle	level	5	71%
	(60%-100%)	(0%-40%)	2	29%
	100%	0%	0	0%
32_Siesta	0%	0%	0	0%
10	0%	100%	1	10%
	(0%-40%)	(60%-100%)	2	20%
	middle	level	4	40%
	(60%-100%)	(0%-40%)	3	30%
	100%	0%	0	0%

Flatliners. Mostly zero change. -- the rest, equally split, mostly monothematically, in a maintenance only, and, mostly expansion.
Expansion is mostly achieved via attribute injections. Maintenance is mostly manifested via type updates.

Radical Sign. Archetype “build; few changes; freeze” – most populous of all. Radical sign change is predominantly achieved via table insertions and deletions. 1/3 of the projects are maintenance oriented; the rest, towards expansion, too often monothematic.

Sigmoid. 8 out of 19 projects have zero change; the rest have mostly low change, mostly monothematic (5 /12). Projects are slightly inclined towards expansion, mostly via attribute injections. There is no maintenance really, except for two large projects with too many table deletions.

Late Riser. projects with small change (with 1 exception), heavily monothematic (9/12). The pattern is slightly biased towards expansion. Maintenance: 4 projects inclined to maintenance, mostly via type updates. The rest: (small volume) expansion-only projects, mostly via attribute injections. .

Quantum Steps. 4/23 projects that are maintenance-only, with change manifested via type updates. When maintenance dominates: mostly via update-type. Mixed nature projects: maintenance is mostly achieved via deletion of entire tables. When expansion is predominant (12/23): via new table births.

Regularly Curated. Change rises to significant volumes. Heavily mixed (only 2 rows in the table): almost all categories (exc. PK) appearing in almost all projects. The mix of change is mostly balanced, slightly in favor of expansions. Expansion is mostly achieved via new tables. Injections clearly exist but significantly less than attributes been born with new tables. Maintenance is mostly achieved via table deletions, with (i) attributes being ejected and (ii) data types being updated constituting a second tier of types in terms of popularity..

Smoking Funnel. Similar to the one of Regularly Curated projects.

Siesta. Heavily mixed, with each project having a large number of change types (at least 3). In contrast to the other categories, siesta projects are slightly in favor of maintenance. Maintenance is split half and half between (a) table deletions and (b) data type updates. Expansion is mostly achieved via table insertions.

Summary of Schema Change Types

- With the exception of few radical sign projects that are oriented towards maintenance, the “Be Quick or Be Dead” family involves small change, frequently being zero, and an inclination towards expansion. Due to the small volume of change, the patterns of the family are frequently monothematic in their internal breakdown of change types.
- In contrast, the rest of the time-related patterns come with higher volumes of change, which is also related to a variety of change types. Both expansion and maintenance are performed with the granule of change being mostly the entire table (being inserted or deleted), rather than the internal restructuring of existing tables.

Lessons Learned and Open Issues

… and why it matters…

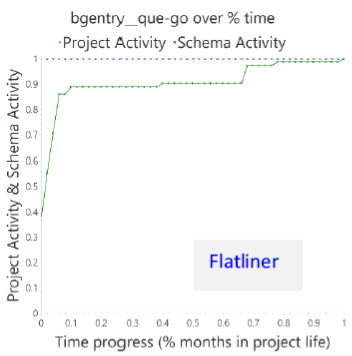
Main contribution

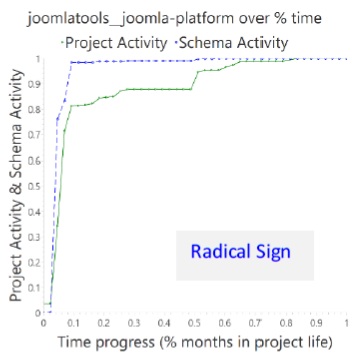
- **For the first time in the literature we know how schema evolution unfolds over time**

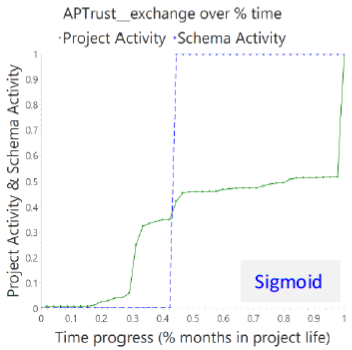
- We have extracted …
- … 8 Patterns of Change in Time…
- … organized in 3 Families

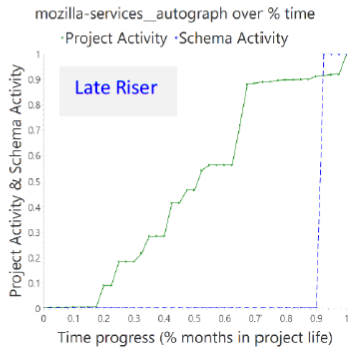
- We verified the results of pattern extraction wrt common sense, generalization, disjointness, cohesion, completeness

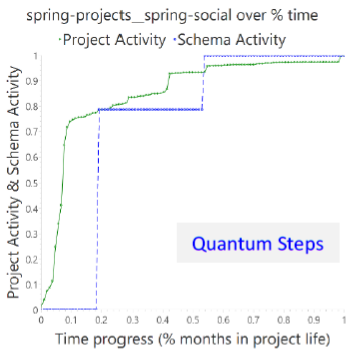
	Count of prjs
1_BeQuickOrBeDead	**97**
11_FlatLiner	23
12_RadicalSign	41
13_Sigmoid	19
14_LateRiser	14
2_StairwayToHeaven	**37**
21_QuantumSteps	23
22_RegularlyCurated	14
3_ScaredToFallAsleepAgain	**17**
31_SmokingFunnel	7
32_Siesta	10
Grand Total	**151**

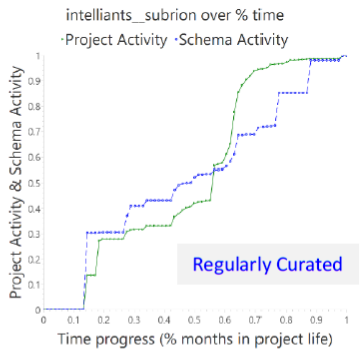


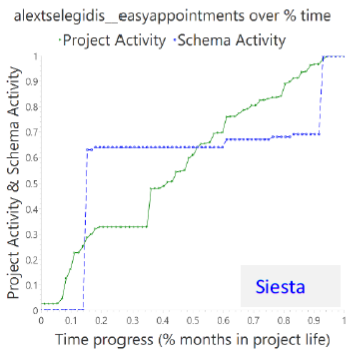


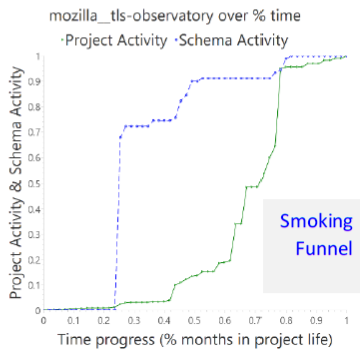






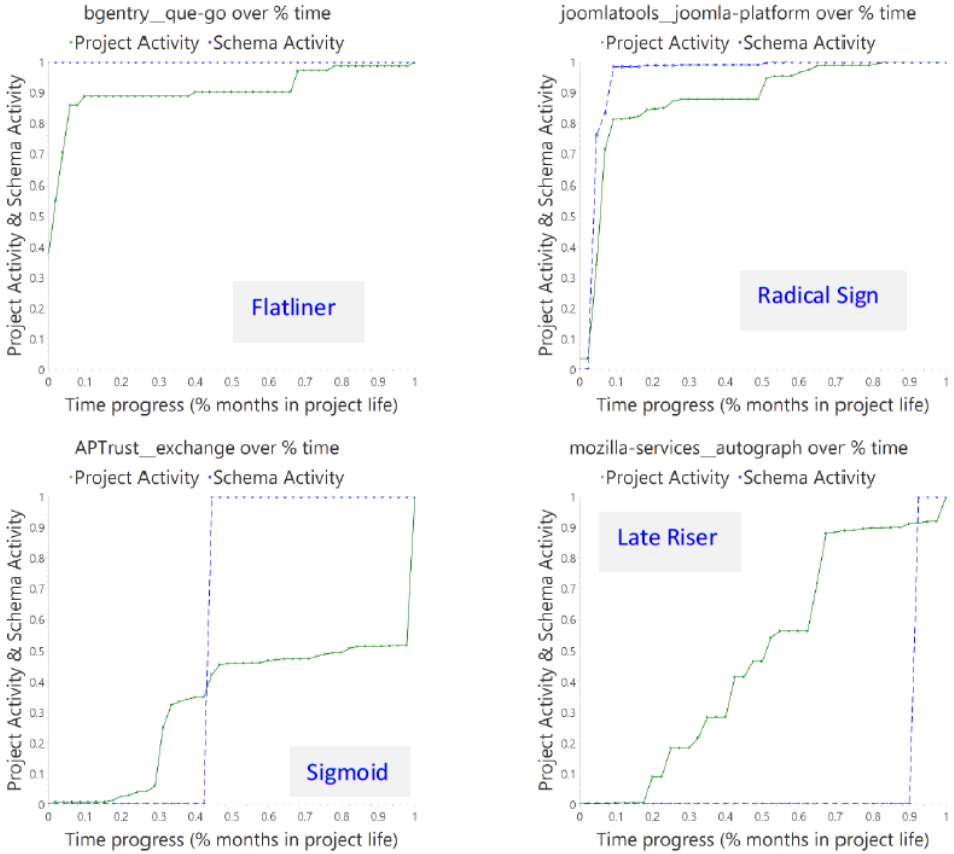






Aversion to change

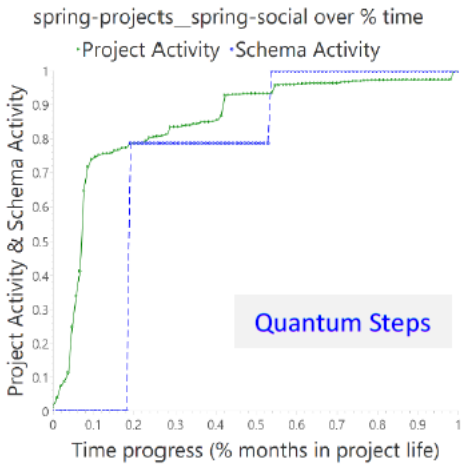
- Almost 2/3 of the corpus, 97/151 evolve with very focused change very close to the point of schema birth - the only difference of the involved patterns is when schema birth takes place.
- This is **in sync with our recent previous findings over a very large corpus of projects**, where 70% of 327 projects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

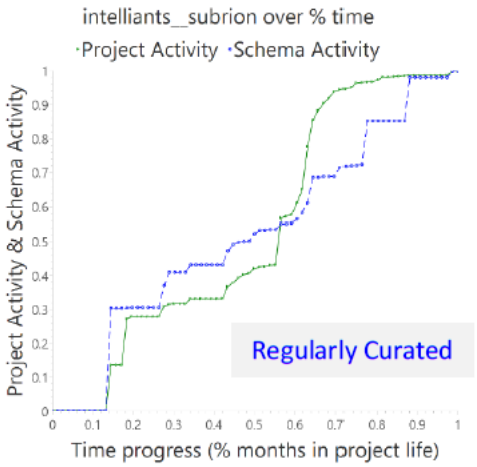


- This is in contrast with all the past research till the 20's, involving small studies, of few, carefully picked prj's that actually showed any change whatsoever

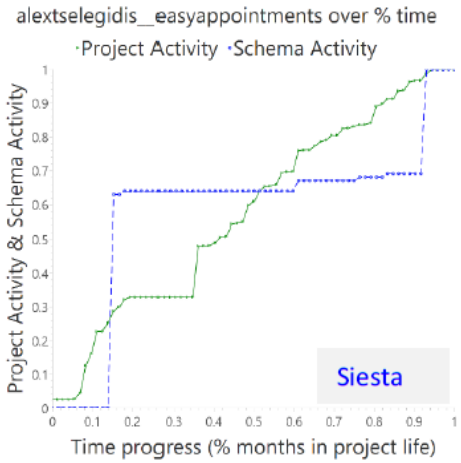
Still, change exists!

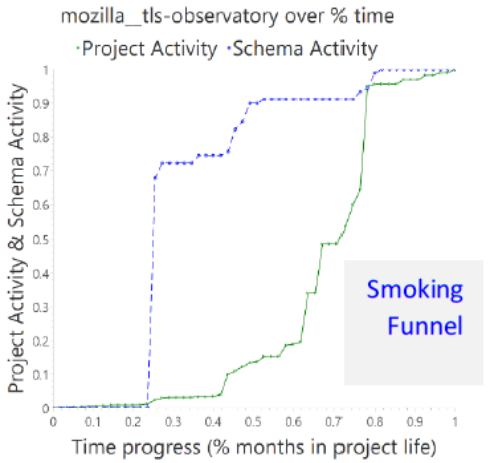
- Almost 25% (37/151) projects evolve with regular change steps, distributed across time, progressively climbing to the top-band over a long period of time.





- 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.

Summary of Schema Change Types

- With the exception of few radical sign projects that are oriented towards maintenance, the “Be Quick or Be Dead” family involves small change, frequently being zero, and an inclination towards expansion. Due to the small volume of change, the patterns of the family are frequently monothematic in their internal breakdown of change types.
- In contrast, the rest of the time-related patterns come with higher volumes of change, which is also related to a variety of change types. Both expansion and maintenance are performed with the granule of change being mostly the entire table (being inserted or deleted), rather than the internal restructuring of existing tables.

Summary of how patterns differ wrt evolution characteristics

- Almost all patterns have median values of the Project Update Period that are very close, i.e., their evolution does NOT relate to the duration of the project
- Almost all patterns have similar volumes of new-born schemata, except for Regularly Curated and Smoking Funnel that have significantly larger mean values of birth volume
- Total Schema Activity (as well as Total Schema Expansion and Total Schema Maintenance which typically follow it) is rather indifferent to the patterns, with small change values except for Regularly Curated and Smoking Funnel who have larger values than the rest.
- Thus, Smoking Funnel and Regularly Curated projects start bigger and contain higher amounts of schema evolution activity than the rest of the projects who start small and typically show lower values of change.

When are schemata born within a project's life?

- 34% of the schemata are born in M0
- 60% of the schemata are born in the first 6 months
- 68% of the schemata are born 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%

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

Born in …	% chance to end up as Be quick or be dead	% chance to end up with some kind of regular change
M0	75%	25%
[M01 – M06]	53%	47%
[M07 – M12]	53%	47%
>M12	64%	36%

	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%

Lessons learned and who cares

- The "freeze-and-build" aversion to change is indeed present at large and concerns 2/3 of the corpus (Be Quick or Be Dead family).
  - For researchers: we need to reflect on our schema design and software development premises.
  - For Developers: To battle aversion to change, development teams should chart the mapping of schema to source code in order to be able to adapt them in sync.
- The rest of the projects evolve differently in time: rare but regular change; densely regular change; and, surprisingly, late change too. Thus, we cannot rely solely on the premise of aversion to change.
- We also have insights on how likely it is to encounter change or rigidity in the life of a schema.
  - Project managers can reserve time up-front for handling change and its impact.
- Sadly atypically for our data engineering community, we contribute with the enrichment of our knowledge –in a principled manner– on how the artifacts that we invent (here: relational databases) are actually used in practice. Part of our contribution is the topic itself, along with the nomenclature, measures and methodological principles used.

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 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).
- (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!

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

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

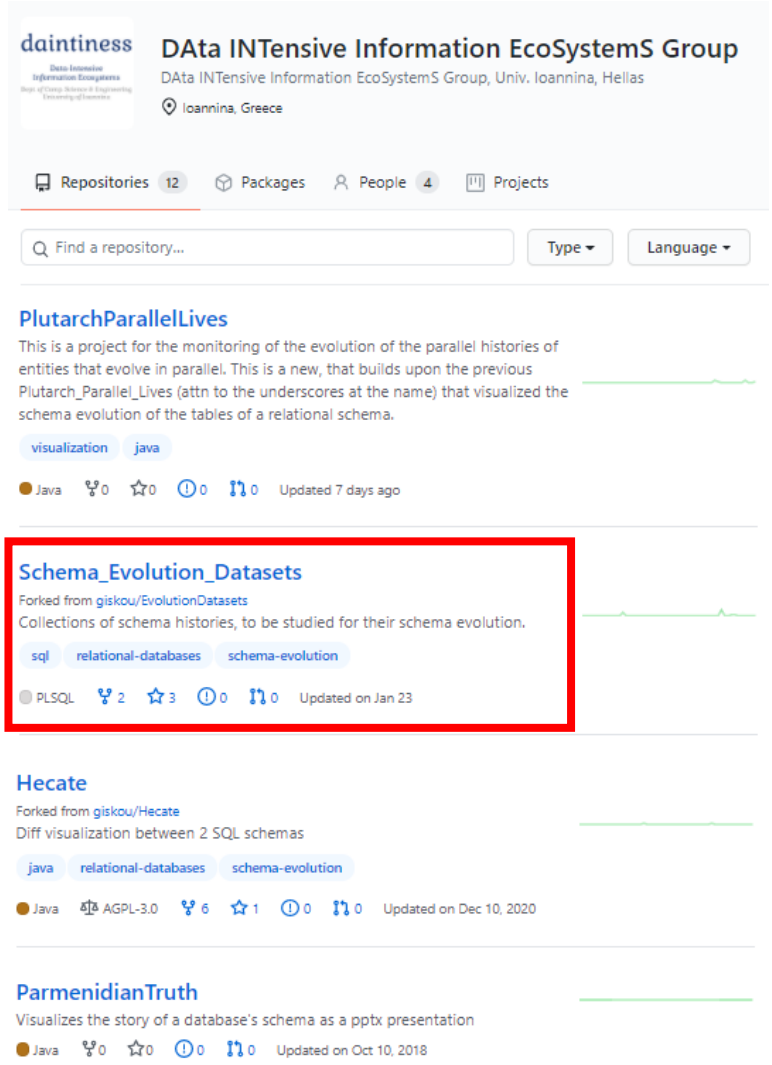
and Code

… 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/