

Extraction Report: 06_technical_narrative

Property	Value
Domain	Narrative
Format	.html
Text length	1,826 chars
Sentences	14

Entities

Spacy NER: 22 entities (222ms)

Entity	Label	Score
The Evolution of Version Control Systems	ORG	
three	CARDINAL	
first	ORDINAL	
only one	CARDINAL	
second	ORDINAL	
CVS	ORG	
2004	DATE	
Apache	FAC	
36 hours	TIME	
third	ORDINAL	
Git and Mercurial	ORG	
Git	PERSON	
Linus Torvalds	ORG	
2005	DATE	
approximately 95%	PERCENT	
GitHub	ORG	
GitLab	ORG	
Bitbucket	ORG	

Gliner NER: 54 entities (217ms)

Entity	Label	Score
Version Control Systems	system_component	0.48
Version Control Systems	algorithm	0.46
Version control systems	system_component	0.60
phases	temporal_concept	0.76
software teams	role	0.80
collaborate	role	0.37
local-only systems	architecture_pattern	0.78
RCS	system_component	0.57
individual files	system_component	0.60
single machine	system_component	0.44
inherent limitation	constraint	0.44
developer	role	0.74
file	system_component	0.47
centralized systems	system_component	0.64
CVS	system_component	0.59
Subversion	protocol	0.60
central server	system_component	0.89

Entity	Label	Score
canonical repository	system_component	0.78
developers	role	0.60
model	algorithm	0.44
team collaboration	role	0.63
single point of failure	constraint	0.75
central		

server | system_component | 0.90 | | down | event_type | 0.39 | | development activity | action | 0.59 | | 2004 | temporal_concept | 0.52 | | Subversion server | system_component | 0.65 | | outage | event_type | 0.43 | | contributions | action | 0.40 | | 36 hours | temporal_concept | 0.89 |

Stanza NER: 23 entities (676ms)

Entity	Label	Score
three	CARDINAL	
first	ORDINAL	
RCS	ORG	
only one	CARDINAL	
second	ORDINAL	
CVS	ORG	
2004	DATE	
Apache	PRODUCT	
36 hours	TIME	
third	ORDINAL	
Git	ORG	
Mercurial	ORG	
Linus Torvalds	PERSON	
2005	DATE	
Today	DATE	
approximately 95%	PERCENT	
GitHub	PRODUCT	
GitLab	PRODUCT	
Bitbucket	ORG	
Git's	ORG	

Flair NER: 14 entities (90ms)

Entity	Label	Score
RCS	MISC	1.00
CVS	MISC	0.55
Subversion	MISC	0.76
Apache	ORG	1.00
Git	MISC	1.00
Mercurial	MISC	1.00
Git	ORG	1.00
Linus Torvalds	PER	1.00
GitHub	MISC	1.00
GitLab	MISC	1.00
Bitbucket	MISC	1.00

Relations (SRL)

28 SRL frames (53ms)

Verb	Agent	Patient	Other
undergo	Evolution of Version Control Systems		

| three major evolutionary phases | | collaborate | software teams | | | consist | first generation | | ARG2=of local
- only systems like RCS , | | track | which | changes to individual files | ARGM-LOC=on a single machine | | be |
systems | simple | | | create | | inherent limitation | | | work | only one developer | These systems were simple but
cr | ARGM-MOD=could, ARGM-LOC=on a file | | introduce | second generation | centralized systems , most
notably CVS a | | | maintain | central server | canonical repository | | | check | developers | working copies | | |
enable | model | team collaboration | | | introduce | | single point of failure | | | go | central
server | | | | stop | all development activity | | ARGM-ADV=When the central
server | | demonstrate | 2004 Subversion server outage at Apache | vulnerability , | ARGM-ADV=halting contribu-
tions for 36 ho | | bring | third generation | distributed version control , led by Git | | | maintain | Every
developer | complete copy of repository | ARGM-ADV=enabling offline work and elimi | | enable | | offline work |
| | eliminate | | single point of failure | | | design | | | ARGM-LOC=in 2005 | | introduce | Git 's branching model
, designed by Lin | concept of cheap branches that fundament | | | change | that | development workflows | | | use
| merge algorithm in Git | three - way merge strategy that resolves | | | resolve | that | conflicts | | | compare | |
current branch , target branch , and the | |

Enriched Extraction (batch-enrich)

Single GPU call: 54 entities, 14 roles, 28 SRL frames (427ms)

Latency Comparison

Method	Latency
Separate (GLiNER 217ms + GLiClass 183ms + SRL 53ms)	453ms
batch-enrich (unified)	427ms
Speedup	1.1x

Per-Sentence Enriched Results (sample)

Sentence	Entities	Role	SRL Frames
The Evolution of Version Control Systems			

The Evo | 6 | Enabler (0.53) | 2 | | The first generation consisted of local-only systems like RC | 4 | State Change
(0.64) | 2 | | These systems were simple but
created an inherent limi | 3 | Condition (0.27) | 3 | | The second generation introduced centralized systems, most
n | 3 | Condition (0.16) | 1 | | A central server maintained the canonical repository, and de | 3 | Variable (0.18) | 2
| | This model enabled team collaboration but introduced a singl | 3 | Enabler (0.88) | 2 | | When the central
server went down, all development act | 3 | Condition (0.25) | 2 | | The 2004 Subversion server outage at Apache
demonstrat | 5 | Effect (0.30) | 1 | | The third generation brought distributed version control, le | 2 | Enabler (0.20)
| 1 | | Every
developer maintains a complete copy of the repos | 4 | Enabler (0.60) | 3 | | Git's branching model, designed by
Linus Torvalds in 2005,

| 4 | Enabler (0.83) | 3 | | The merge algorithm in Git uses a three-way merge strategy t | 5 | Condition (0.39) | 3 |
 | Today, Git dominates with approximately 95% market share amo | 3 | Enabler (0.23) | 1 | | The shift from
 centralized to distributed systems trig | 6 | Enabler (0.38) | 2 |

Entity Type Distribution (enriched)

Label	Count
system_component	17
role	7
algorithm	5
protocol	4
data_structure	4
temporal_concept	3
architecture_pattern	3
constraint	3
action	3
event_type	2
metric	2
cognitive_concept	1

Role Distribution (enriched)

Role	Count
Enabler (Causal)	7
Condition (State)	4
State Change (Other)	1
Variable (Scientific)	1
Effect (Event)	1

QLang Sentences

Gliclass: 14 classifications (183ms)

Causal (7)

- [Enabler] (0.53) The Evolution of Version Control Systems

The Evolution of Version Co...

- [Enabler] (0.88) This model enabled team collaboration but introduced a single point of failure....
- [Enabler] (0.20) The third generation brought distributed version control, led by Git and Mercuri...
- [Enabler] (0.60) Every

developer maintains a complete copy of the repository, enabling offl...

- [Enabler] (0.83) Git's branching model, designed by Linus Torvalds in 2005,

introduced the ...

- [Enabler] (0.23) Today, Git dominates with approximately 95% market share among developers....
- [Enabler] (0.38) The shift from

centralized to distributed systems triggered the emergence ...

Event (1)

- [Effect] (0.30) The 2004 Subversion server outage at Apache

demonstrated this vulnerabilit...

Other (1)

- [State Change] (0.64) The first generation consisted of local-only systems like RCS,

which track...

Scientific (1)

- [Variable] (0.18) A central server maintained the canonical repository, and developers checked out...

State (4)

- [Condition] (0.27) These systems were simple but

created an inherent limitation: only one dev...

- [Condition] (0.16) The second generation introduced centralized systems, most notably CVS and later...
- [Condition] (0.25) When the central

server went down, all development activity stopped....

- [Condition] (0.39) The merge algorithm in Git uses a three-way merge strategy that resolves conflic...

Qualtron: 3 classifications (5660ms)

Core (1)

- [Concept] (0.95) Git's branching model, designed by Linus Torvalds in 2005, introduced the concep...

Event (1)

- [Effect] (0.95) This model enabled team collaboration but introduced a single point of failure....

Other (1)

- [Title] (0.95) The Evolution of Version Control Systems...

QHG Process Models

FSM (1111ms)

Could not parse structured output

None

BPMN (101ms)

Could not parse structured output

```
{"elements": [], "flows": []}
```

DFG (2741ms)

Could not parse structured output

None

KnowledgeState (1881ms)

Could not parse structured output

None

CNL / QNR2 Rules

0 rule patterns detected

Heuristic Facts & Rules

2 facts, 1 rules

Facts (sample)

- demonstrated this vulnerability, halting contributions for 36 hours....
- Today, Git dominates with approximately 95% market share among developers. The shift from...

Rules (sample)

- [conditional] This model enabled team collaboration but introduced a single point of failure. When the central...

Topics (Gensim LDA)

- **Topic 0:** {'word': 'generation', 'weight': 0.05}, {'word': 'git', 'weight': 0.0273}, {'word': 'centralized', 'weight': 0.0273}, {'word': 'subversion', 'weight': 0.0273}, {'word': 'systems', 'weight': 0.0273}, {'word': 'version', 'weight': 0.0273}, {'word': 'introduced', 'weight': 0.0273}, {'word': 'control', 'weight': 0.0273}
- **Topic 1:** {'word': 'server', 'weight': 0.0444}, {'word': 'systems', 'weight': 0.0306}, {'word': 'distributed', 'weight': 0.0306}, {'word': 'developers', 'weight': 0.0167}, {'word': 'development', 'weight': 0.0167}, {'word': 'central', 'weight': 0.0167}, {'word': 'like', 'weight': 0.0167}, {'word': 'collaboration', 'weight': 0.0167}
- **Topic 2:** {'word': 'this', 'weight': 0.0388}, {'word': 'contributions', 'weight': 0.0387}, {'word': 'hours', 'weight': 0.0387}, {'word': 'vulnerability', 'weight': 0.0387}, {'word': 'halting', 'weight': 0.0387}, {'word': 'demonstrated', 'weight': 0.0387}, {'word': 'git', 'weight': 0.0065}, {'word': 'systems', 'weight': 0.0065}
- **Topic 3:** {'word': 'merge', 'weight': 0.0478}, {'word': 'developer', 'weight': 0.0261}, {'word': 'work', 'weight': 0.0261}, {'word': 'repository', 'weight': 0.0261}, {'word': 'that', 'weight': 0.0261}, {'word': 'git', 'weight': 0.0261}, {'word': 'copy', 'weight': 0.0261}, {'word': 'eliminating', 'weight': 0.0261}
- **Topic 4:** {'word': 'single', 'weight': 0.033}, {'word': 'model', 'weight': 0.0227}, {'word': 'point', 'weight': 0.0227}, {'word': 'failure', 'weight': 0.0227}, {'word': 'branch', 'weight': 0.0227}, {'word': 'introduced', 'weight': 0.0227}, {'word': 'git', 'weight': 0.0227}, {'word': 'systems', 'weight': 0.0227}

Summary (Sumy LexRank)

The second generation introduced centralized systems, most notably CVS and later Subversion. A central server maintained the canonical repository, and developers checked out working copies. This model enabled team collaboration but introduced a single point of failure. The third generation brought distributed version control, led by Git and Mercurial. Git's branching model, designed by Linus Torvalds in 2005,