

# SituationReport

## testdata\_generator

### Manual de Utilizator

Generați date sintetice de probleme Jira pentru SituationReport

---

situation-report -- [github.com/Jaegerfeld/situation-report](https://github.com/Jaegerfeld/situation-report)

Pentru Agile Coaches si Manageri PI -- Version 0.19.0

# 1 Ce este Generatorul de Date de Test?

---

Generatorul de Date de Test (Testdata Generator) creează date sintetice de probleme Jira în formatul Jira REST API. Fișierele generate pot fi procesate direct de modulul transform\_data — fără o instanță Jira reală.

Generatorul simulează istorii realiste de flux de lucru: problemele parcurg etapele definite, petrec perioade variabile în fiecare etapă, revin ocazional și se închid la o rată configurabilă.

Cazuri de utilizare tipice:

- Testarea unei instalări noi: Verificați dacă transform\_data și build\_reports funcționează corect
- Demo-uri și prezentări: Date realiste fără preocupări de confidențialitate
- Training: Mediu de învățare controlat cu date cunoscute
- Dezvoltare: Date de test reproductibile prin parametrul seed

Generatorul completează modulul Get Data (în curs de dezvoltare), care încarcă date reale direct din Jira. Pentru exportul manual al datelor reale Jira, consultați Manualul de Utilizator Get Data.

## 2 Cerințe prealabile și pornire

### 2.1 Pachet portabil (recomandat)

Pachetul portabil include deja Python integrat — nu este necesară o instalare separată.

| Sistem de operare | Fișier de pornire                                   |
|-------------------|-----------------------------------------------------|
| Windows           | TestdataGenerator.bat (dublu-clic)                  |
| macOS             | TestdataGenerator.command (clic dreapta → Deschide) |
| Linux             | ./TestdataGenerator.sh                              |

*Notă pentru Windows: Dacă la pornire apare o eroare privind Python, fișierul ZIP a fost probabil blocat de Windows. Soluție: clic dreapta pe ZIP → Proprietăți → Securitate → Deblocare → OK → reextragere.*

### 2.2 Din codul sursă (necesită Python)

```
python -m testdata_generator
```

### 2.3 Interfața utilizator

La pornire se deschide fereastra principală cu câmpuri de intrare pentru toți parametrii. Singurul câmp obligatoriu este fișierul de flux de lucru — toate celelalte au valori implicite sensibile.

testdata\_generator 0.9.9

Hilfe

Workflow-Datei

Ausgabedatei (JSON)

Projekt-Key: TEST

Anzahl Issues: 100

Von (YYYY-MM-DD): 2025-01-01

Bis (YYYY-MM-DD): 2025-12-31

Issue-Typen (Typ:Gewicht ...)

Completion-Rate (0-1): 0.7

To-Do-Rate (0-1): 0.15

Backflow-Prob. (0-1): 0.1

Seed (leer = zufällig)

Ø Cycle Time (Tage, leer=auto)

Std.-Abw. (Tage, leer=auto)

Muster: ☒ Kein Muster ☐ Triangle ☐ Flat Triangle ☐ Cluster ☐ Batch

PI-Dauer (Wochen): 12

Generieren

Log

Fig. 1: Interfața utilizator a Generatorului de Date de Test

## 3 Fișierul de flux de lucru

Fișierul de flux de lucru definește etapele tablei Jira și este singurul parametru obligatoriu. Folosește același format ca `transform_data` — un singur fișier funcționează pentru ambele module.

### 3.1 Format

Fiecare linie descrie o etapă. Aliasurile (denumiri alternative de stare în exportul Jira) sunt separate prin două puncte. Două linii speciale marchează începutul și sfârșitul lucrului activ:

```
NumeCanonic:Alias1:Alias2
<First>PrimaEtapaActiva
<Closed>EtapaInchisa
```

| Linie                     | Semnificație                                                    |
|---------------------------|-----------------------------------------------------------------|
| NumeCanonic:Alias1:Alias2 | Etapă cu unul sau mai multe denumiri alternative Jira           |
| NumeEtapa (fără :)        | Etapă fără aliasuri                                             |
| NumeEtapa                 | Prima etapă activă — unde începe prelucrarea (start cycle time) |
| NumeEtapa                 | Etapă închisă — marchează o problemă ca finalizată              |

### 3.2 Exemplu de flux ART\_A

Exemplul inclus `workflow_ART_A.txt` modelează un flux SAFe ART tipic.

```
Funnel:New:Open:To Do
Analysis:In Analysis:Estimated
Program Backlog:Ready for Dev:Backlog
Implementation:In Implementation:In Review:In Progress
Blocker
Validating on Staging:QA
Deploying to Production:Ready for Development
Releasing:Completed
Done:Canceled
<First>Analysis
<Closed>Releasing
```

## 4 Referință parametri

Toți parametrii sunt disponibili atât în interfața grafică, cât și în linia de comandă.

| Parametru                | Implicit                           | Descriere                                                           |
|--------------------------|------------------------------------|---------------------------------------------------------------------|
| --workflow FILE          | (obligatoriu)                      | Fișier de definiție a fluxului de lucru (.txt)                      |
| --output FILE.json       | _generated.json                    | Calea fișierului de ieșire                                          |
| --project KEY            | TEST                               | Cheia proiectului Jira pentru problemele generate                   |
| --issues N               | 100                                | Numărul de probleme de generat                                      |
| --from-date YYYY-MM-DD   | 2025-01-01                         | Data de creare cea mai timpurie                                     |
| --to-date YYYY-MM-DD     | 2025-12-31                         | Data de tranziție cea mai târzie                                    |
| --issue-types TYPE:W ... | Feature:0.6 Bug:0.3<br>Enabler:0.1 | Tipuri de probleme cu ponderi                                       |
| --completion-rate FLOAT  | 0.7                                | Proporția problemelor care ajung la etapa închisă (0–1)             |
| --todo-rate FLOAT        | 0.15                               | Proporția problemelor deschise care rămân în etapele timpurii (0–1) |
| --backflow-prob FLOAT    | 0.1                                | Probabilitatea unei tranziții înapoi la fiecare pas (0–1)           |
| --seed INT               | (aleatoriu)                        | Seed pentru rezultate reproductibile (opțional)                     |
| --mean-cycle-days FLOAT  | (niciunul)                         | Cycle time mediu în zile (lognormal); activează controlul CT        |
| --std-cycle-days FLOAT   | (30 % din medie)                   | Abaterea standard a cycle time în zile                              |
| --pattern TIPAR          | none                               | Anti-pattern: none / triangle / flat_triangle / cluster / batch     |
| --pi-duration-weeks INT  | 12                                 | Durata ciclului PI în săptămâni (pentru cluster/batch)              |
| --pattern-strength 0-100 | 50                                 | Intensitatea modelului 0–100 (0=subtil, 50=implicit, 100=puternic)  |

## 4b Modele de Anti-Pattern de flux

Începând cu versiunea 0.9.9, Generatorul de Date de Test poate simula anti-pattern-uri tipice de flux din literatura lui Vacanti și Singh. Combinați `--mean-cycle-days` cu `--pattern`.

| Tipar         | Aspect scatter plot                         | Descriere                                                                                                                                                                                  |
|---------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| none          | Nor de puncte uniform                       | Cycle time aleatoriu fără tendință — comportament implicit                                                                                                                                 |
| triangle      | Triunghi: unghi drept jos-dreapta           | Cycle time crește liniar în timp (multiplicator $\sim 1-4\times$ la intensitate implicită, mai mare cu <code>--pattern-strength</code> ). Indică un proces care devine progresiv mai lent. |
| flat_triangle | Triunghi, creștere atenuată la final        | Ca triangle, dar creșterea este atenuată prin $\tanh(2.5\times t)$ . Arată o deteriorare a procesului care se stabilizează.                                                                |
| cluster       | Grupuri verticale de puncte la finalul PI   | Cycle time rămâne stabil, dar majoritatea problemelor sunt livrate în ultimele 2 săptămâni înainte de finalul PI. Comportament determinat de termen limită.                                |
| batch         | Coloane de înălțime variabilă la finalul PI | Ca cluster, dar cu cycle time foarte variabilă ( $0.1-3\times$ medie). Arată că problemele scurte și lungi sunt amânate împreună.                                                          |

### Controlul cycle time

```
# Tipar triunghi: CT crește de la 20z la 80z
python -m testdata_generator \
  --workflow workflow.txt --issues 300 \
  --pattern triangle \
  --mean-cycle-days 20 --std-cycle-days 6 \
  --completion-rate 0.8 --seed 1 \
  --output triangle.json
```

## 5 ART\_A – Exemplu complet

---

Acest exemplu arată întregul pipeline de la generator la raportul final, folosind proiectul fictiv ART\_A.

### Pasul 1: Generare date de test

```
python -m testdata_generator \
    --workflow testdata_generator/workflow_ART_A.txt \
    --project ART_A --issues 200 \
    --from-date 2025-01-01 --to-date 2025-12-31 \
    --completion-rate 0.75 --seed 42 \
    --output ART_A_generated.json
```

### Pasul 2: Procesare cu transform\_data

```
python -m transform_data ART_A_generated.json \
    --workflow testdata_generator/workflow_ART_A.txt
```

### Pasul 3: Creare raport cu build\_reports

```
python -m build_reports
```



## 5b Scenariul de portofoliu

Cu `--scenario portfolio` generatorul creează într-un singur pas un portofoliu demo complet: două soluții cu câte trei ART-uri, inclusiv toate artefactele lanțului de procesare.

În GUI: secțiunea 'Demo portfolio' din partea de jos a ferestrei generează scenariul într-un director la alegere printr-un clic (câmpul seed este preluat, implicit 42); 'Open Portfolio Report' redă raportul de portofoliu și îl deschide în browser — fără linie de comandă.

```
python -m testdata_generator --scenario portfolio \
    --output demo/ --seed 42
```

Directorul țintă conține apoi:

| Artefact            | Conținut                                                 |
|---------------------|----------------------------------------------------------|
| workflows/          | două fișiere de flux de lucru (etape Alpha și Beta)      |
| raw/                | șase exporturi JSON Jira (o rulare de generator per ART) |
| arts/               | fișiere IssueTimes, CFD și Transitions per ART           |
| solution_alpha.json | config de soluție fără stage_map (cale implicită)        |
| solution_beta.json  | config de soluție cu stage_map propriu (schema 2)        |
| risks_*.json        | registru de riscuri ROAM per soluție                     |
| nfr_*.json          | registru NFR/runway per soluție                          |
| capabilities_*.json | capability map per soluție                               |
| dependencies_*.json | registru de dependențe per soluție                       |
| decisions_*.json    | log de decizii/presupuneri per soluție                   |
| portfolio.json      | config de portofoliu peste ambele soluții                |
| pi_config.json      | intervale PI pe fereastra de date                        |
| README.md           | descrierea poveștilor încorporate                        |

Datele sunt plasate relativ la data generării, astfel încât semaforul de calitate al raportului de portofoliu să le evalueze ca actuale. Trei povești sunt încorporate:

Povești încorporate:

- ART Alpha-3 este valoarea aberantă (cycle time de circa trei ori mai mare) — evidențiat cu roșu în raportul de comparație al Solution Alpha.
- ART Beta-3 livrează date slabe (fără CFD, puține issue-uri începute, date vechi de 60 de zile) — încredere 'low' în tabelul de calitate.
- Solution Beta agregă printr-un stage\_map propriu (Vorlauf/Umsetzung/Fertig); Solution Alpha folosește calea implicită.
- Board ROAM: două riscuri owned sunt intenționat vechi (45/50 de zile) — evidențierea aging se activează.
- NFR & runway: NFR-ul API al soluției Beta este încălcat, iar un element runway este o lacună întârziată — dashboard-ul arată roșu.
- Capability map: capacitatea data-insights a soluției Beta este critică, o capacitate Alpha nu are ART (uncovered).
- Heatmap de dependențe: Alpha-1 → Alpha-3 blocată și întârziată; Beta-1 → Alpha-1 ca integrare cross-solution.
- Log de decizii: presupunerea deschisă a soluției Beta și-a depășit data de revizuire — marcată roșu 'review due'.

Directorul este direct utilizabil: `python -m portfolio demo/portfolio.json` construiește raportul de portofoliu; configurile de soluție funcționează și individual. Același seed produce date identice în aceeași zi.

## 6 Întrebări frecvente (FAQ)

---

### Diferența dintre datele de test și datele reale?

Pentru demo-uri, training și teste de instalare, datele de test sunt alegerea mai bună — fără preocupări de confidențialitate, reproductibile și complet controlate. Pentru analize reale de flux, datele reale reflectă starea actuală a fluxului de lucru.

*Pentru exportul manual al datelor reale Jira: Consultați Manualul de Utilizator Get Data.*

## 7 Glosar

| Termen     | Explicație                                                                      |
|------------|---------------------------------------------------------------------------------|
| ART        | Agile Release Train — structură de echipă multiplă în SAFe                      |
| API        | Application Programming Interface — interfață de programare                     |
| CFD        | Cumulative Flow Diagram — număr cumulat de probleme per etapă                   |
| Cycle Time | Timp de la prima etapă activă până la etapa închisă                             |
| Issue      | Element de lucru în Jira (Feature, Bug, Enabler, Story, ...)                    |
| JSON       | JavaScript Object Notation — format exporturi API Jira                          |
| Seed       | Valoare de start pentru generatorul aleatoriu — același seed = același rezultat |
| Stage      | Un pas în fluxul de lucru (corespunde unei coloane Jira)                        |
| Workflow   | Secvență ordonată de etape prin care trece o problemă                           |