

Automatizare de raportare cu dry-run

Fișă publică de proiect · exerciții și criterii de acceptare

Pornești de la un raport manual și construiești un proces repetabil pentru fișiere și date. Trimiterea emailurilor rămâne în mediu de test.

Livrabile

- Plan de automatizare
- Dry-run
- Raport CSV
- Jurnal
- Programare

Criterii de acceptare

- Căile sunt validate
- Coliziunile opresc operația
- Rulările simultane sunt prevenite

Decizie de automatizare cu cost de întreținere

Un script merită construit când procesul este repetabil și verificabil, nu doar enervant. Estimează timpul economisit, timpul de dezvoltare și corecturile. O operație rară cu risc mare poate fi mai potrivită pentru un checklist decât pentru execuție automată. Începe cu un raport care nu modifică sursa, apoi adaugă scrierea numai după validare. Păstrează un mod de simulare pentru operațiile pe fișiere sau date.

Exemplu de lucru

Ipoteze: operație manuală 20 minute/săptămână; verificare script 5 minute; dezvoltare 180 minute.

Economie: 15 minute/săptămână.

Recuperare teoretică: 12 săptămâni, fără întreținere.

Pilot: numai raport, fără mutări sau ștergeri.

Acceptare: rezultate identice pe trei seturi de test.

Interpretarea rezultatului

Estimarea nu este promisiune comercială. Dacă apare săptămânal o corectură de 15 minute, economia dispare. Include cazul cu intrare invalidă și schimbarea formatului, fiindcă acestea determină costul de întreținere.

Exercițiu de consolidare

Compară scriptul cu un checklist manual îmbunătățit și explică în ce situație acesta din urmă poate fi suficient.

1. Ce merită automatizat

Compară scriptul cu un checklist manual îmbunătățit și explică în ce situație acesta din urmă poate fi suficient.

Verificare orientativă

Estimarea nu este promisiune comercială. Dacă apare săptămânal o corectură de 15 minute, economia dispare. Include cazul cu intrare invalidă și schimbarea formatului, fiindcă acestea determină costul de întreținere.

De evitat

Automatizarea fără criteriu de acceptare nu are rezultat verificabil.

Operații pe fișiere cu simulare și domeniu restrâns

Înainte de redenumire sau mutare, rezolvă căile și confirmă că toate țintele rămân în directorul laboratorului. Construiește mai întâi lista operațiilor și detectează coliziunile. Nu executa o mutare recursivă pe o cale calculată fără verificare. Pentru acest exercițiu, nu șterge fișiere și nu urmărești linkuri simbolice către alte directoare. Un mod dry-run trebuie să arate exact sursa și destinația.

Exemplu de lucru

```
from pathlib import Path
root = Path('laborator-fisiere').resolve()
root.mkdir(exist_ok=True)
for src in sorted(root.glob('*.txt')):
    if src.is_symlink():
        continue
    dest = src.with_name('arhiva-' + src.name)
    if not dest.resolve().is_relative_to(root):
        raise ValueError('Destinație în afara laboratorului')
    if dest.exists():
        raise FileExistsError(dest)
    print('SIMULARE', src.name, '->', dest.name)
```

Interpretarea rezultatului

Scriptul doar afișează planul. Nu adăuga rename până când lista și coliziunile nu sunt verificate pe copii. Repetarea poate genera prefixe multiple dacă nu definești regula pentru fișiere deja procesate. Păstrează un jurnal al operațiilor efectiv aplicate.

Exercițiu de consolidare

Adaugă două fișiere cu destinație conflictuală și verifică faptul că planul este refuzat înaintea oricărei modificări.

2. Operații pe fișiere și foldere

Adaugă două fișiere cu destinație conflictuală și verifică faptul că planul este refuzat înaintea oricărei modificări.

Verificare orientativă

Scriptul doar afișează planul. Nu adăuga rename până când lista și coliziunile nu sunt verificate pe copii. Repetarea poate genera prefixe multiple dacă nu definești regula pentru fișiere deja procesate. Păstrează un jurnal al operațiilor efectiv aplicate.

De evitat

Nu șterge sau muta recursiv pe căi nevalidate.

Agregare CSV cu total de control

Un raport trebuie să permită reconcilierea cu intrarea. Pentru vânzări fictive, agregă sumele în bani după categorie și păstrează numărul de rânduri valide și respinse. Nu converți valori invalide în zero, fiindcă totalul ar părea corect tehnic, dar ar omite operații. Separă conversia de agregare și scrie erorile într-un fișier distinct. Pentru Excel, verifică tipurile celulelor și formulele după biblioteca aleasă.

Exemplu de lucru

```
Intrare fictivă:  
Categorie;Suma_bani  
service;10000  
piese;5000  
service;2500  
Rezultat: service=12500; piese=5000; total=17500.  
Control: suma categoriilor = totalul rândurilor valide.  
Metadata: 3 rânduri valide, 0 respinse, moneda RON.
```

Interpretarea rezultatului

Totalul de 17.500 bani înseamnă 175 lei. Nu amesteca monede diferite în același total fără o regulă de conversie și o sursă documentată. Dacă raportul folosește perioade, definește fusul orar și limitele intervalului.

Exercițiu de consolidare

Adaugă un rând invalid și unul în altă monedă. Raportul trebuie să le trateze explicit, nu să le însumeze fără explicație.

3. Rapoarte din Excel/CSV

Adaugă un rând invalid și unul în altă monedă. Raportul trebuie să le trateze explicit, nu să le însumeze fără explicație.

Verificare orientativă

Totalul de 17.500 bani înseamnă 175 lei. Nu amesteca monede diferite în același total fără o regulă de conversie și o sursă documentată. Dacă raportul folosește perioade, definește fusul orar și limitele intervalului.

De evitat

Ignorarea liniilor greșite poate falsifica raportul.

Email pregătit fără expediere accidentală

Automatizarea emailului trebuie să distingă generarea, aprobarea și expedierea. În laborator, construiește mesajul și salvează-l ca fișier .eml, fără server SMTP și fără destinatari reali. Verifică subiectul, corpul și atașamentele înaintea integrării. Nu pune parole în script și nu scrie adrese private în loguri. Pentru expediere reală, ai nevoie de autentificare, limite, tratarea erorilor și prevenirea duplicării.

Exemplu de lucru

```
from email.message import EmailMessage
from pathlib import Path
msg = EmailMessage()
msg['From'] = 'raport@example.test'
msg['To'] = 'operator@example.test'
msg['Subject'] = 'Raport demonstrativ de stoc'
msg.set_content('Produsul B are stoc zero. Date fictive.')
Path('raport-demo.eml').write_bytes(bytes(msg))
print('Mesaj pregătit, nu expediat.')
```

Interpretarea rezultatului

Fișierul poate fi inspectat într-un editor sau client de mail de laborator. Domeniul example.test este pentru exemple, nu pentru livrare. Un status „pregătit” nu trebuie transformat în „trimis” până când furnizorul nu confirmă acceptarea conform contractului.

Exercițiu de consolidare

Adaugă un atașament CSV fictiv și verifică numele și conținutul fără a conecta un cont real.

4. Emailuri automate

Adaugă un atașament CSV fictiv și verifică numele și conținutul fără a conecta un cont real.

Verificare orientativă

Fișierul poate fi inspectat într-un editor sau client de mail de laborator. Domeniul example.test este pentru exemple, nu pentru livrare. Un status „pregătit” nu trebuie transformat în „trimis” până când furnizorul nu confirmă acceptarea conform contractului.

De evitat

Exercițiul nu necesită trimiterea către clienți reali.

Extragere din HTML local înainte de scraping extern

Începe cu un fișier HTML controlat ca să verifici selectorii și tratarea câmpurilor lipsă. Pentru site-uri externe, verifică permisiunile, termenii, limitele și existența unui API potrivit. Nu ocoli autentificarea sau protecțiile și nu colecta date personale fără un scop și un temei adecvat. Structura HTML se poate schimba, deci scriptul trebuie să detecteze un rezultat neașteptat în loc să producă un raport gol aparent valid.

Exemplu de lucru

```
HTML local: <article class="produs"><h2>Pompă</h2><span class="pret">50 lei</span></article>
Algoritm: parsează documentul; selectează article.produs; citește h2 și span.pret; validează
prezența; păstrează sursa.
Cazuri: produs complet; preț lipsă; selector schimbat; zero produse.
Nu deduce: preț numeric din orice text fără regulă de format.
```

Interpretarea rezultatului

Zero produse poate însemna pagină fără rezultate sau selector stricat. Verifică un marker cunoscut al paginii și raportează diferența. Nu prezenta prețul extras ca ofertă curentă fără dată și context; el este doar observația din răspunsul primit.

Exercițiu de consolidare

Schimbă clasa pret în HTML și verifică faptul că scriptul semnalează lipsa câmpului, nu salvează zero.

5. Scraping de bază

Schimbă clasa pret în HTML și verifică faptul că scriptul semnalează lipsa câmpului, nu salvează zero.

Verificare orientativă

Zero produse poate însemna pagină fără rezultate sau selector stricat. Verifică un marker cunoscut al paginii și raportează diferența. Nu prezenta prețul extras ca ofertă curentă fără dată și context; el este doar observația din răspunsul primit.

De evitat

Nu colecta date personale sau conținut protejat fără temei.

Script programat cu cale, mediu și blocare

Un script care merge în terminal poate eșua programat din cauza directorului curent, mediului virtual sau permisiunilor. Folosește căi absolute și un utilizator cu acces minim. Păstrează logul și codul de ieșire, iar pentru execuții lungi previne suprapunerea. Nu include secrete direct în comanda programată. Exemplul cron este pentru Linux și trebuie adaptat; pe Windows se folosește mecanismul de programare al sistemului.

Exemplu de lucru

Exemplu Linux de laborator:

```
0 8 * * * /cale/proiect/.venv/bin/python /cale/proiect/raport.py
```

În script: căi bazate pe `Path(__file__).resolve().parent`.

Controale: lock sau mecanism echivalent; timeout; log separat; cod de ieșire nenul la eșec.

Test: rulează din alt director și cu același utilizator ca programarea.

Interpretarea rezultatului

Programarea nu garantează că operația a reușit. Verifică rezultatul și notifică operatorul printr-un mecanism autorizat dacă este necesar. Nu marca raportul complet când sursa a eșuat. Definește politica pentru zile ratate și reluări.

Exercițiu de consolidare

Simulează o execuție care durează mai mult decât intervalul și descrie cum eviți două procese care scriu același fișier.

6. Programarea scripturilor

Simulează o execuție care durează mai mult decât intervalul și descrie cum eviți două procese care scriu același fișier.

Verificare orientativă

Programarea nu garantează că operația a reușit. Verifică rezultatul și notifică operatorul printr-un mecanism autorizat dacă este necesar. Nu marca raportul complet când sursa a eșuat. Definește politica pentru zile ratate și reluări.

De evitat

Cron nu are automat mediul terminalului tău interactiv.

Atelier final: Automatizare cu plan înainte de execuție

Timp de lucru estimat: 40 de minute. Folosește datele de laborator de mai jos și salvează rezultatul separat de orice proiect real.

Date de pornire

Director de laborator: raport.csv, raport-copie.csv și note.txt. Operația dorită este copierea fișierelor CSV într-un director separat cu un nume derivat din data raportului. Simulează o coliziune de nume.

1. Pregătire — 5 minute

Extrage intrările și condițiile din enunț. Notează ce rezultat trebuie produs și ce informații nu sunt confirmate. Pregătește fișierele sau tabelul în care vei păstra dovada lucrului.

2. Construire — 15 minute

Construiește întâi un dry-run care enumeră sursa și destinația. Validează căile și oprește execuția la coliziune. Înregistrează fiecare rezultat în jurnal.

3. Variație și verificare — 15 minute

Adaugă un fișier fără permisiune de citire în mediul de test. Decide dacă procesul continuă sau se oprește și arată cum se vede decizia în raport.

4. Autoevaluare — 5 minute

Compară rezultatul cu reperele de mai jos. Înregistrează cel puțin o eroare identificată sau un caz-limită verificat, plus corectura ori explicația comportamentului observat.

Repere de rezolvare

Dry-run nu modifică fișiere. Coliziunea nu suprascrive date. O nouă rulare are comportament documentat, iar erorile nu sunt ascunse într-un mesaj generic de succes.

Ce predai

- Plan de automatizare
- Dry-run
- Raport CSV
- Jurnal
- Programare

Criterii de acceptare

- Căile sunt validate
- Coliziunile opresc operația
- Rulările simultane sunt prevenite

Păstrează o versiune inițială și una revizuită, cu explicația diferențelor. Dacă un criteriu nu este îndeplinit, revino la lecția relevantă și repetă verificarea. Durata este orientativă; instalarea mediului și aprofundarea pot necesita timp suplimentar.

Registrul de lucru

Pentru fiecare etapă notează intrarea, rezultatul, dovada verificării, eroarea găsită și corectura. Nu introduce date personale sau secrete în fișa partajată.

Documentație și surse

- Python Tutorial — Python Software Foundation

<https://docs.python.org/3/tutorial/>