

Integrarea sistemelor informatice



Suport curs nr. 7

Programator >> Arhitect

Procesul de integrare – Etape

2024-2025

Obiective

- Identificarea etapelor de integrare a sistemelor informatice
 - La nivelul de date
 - La nivelul proceselor de afaceri
 - La nivelul interfețelor aplicațiilor
 - Etape adiționale

Recapitulare – probleme de integrare

- Atunci când avem mai multe componente / niveluri arhitecturale este necesară o analiză a **interoperabilității**
 - Probleme de natură **tehnică**
 - Probleme legate de **politica sistemului**
 - Probleme legate de **logistică**



Image by Freepik

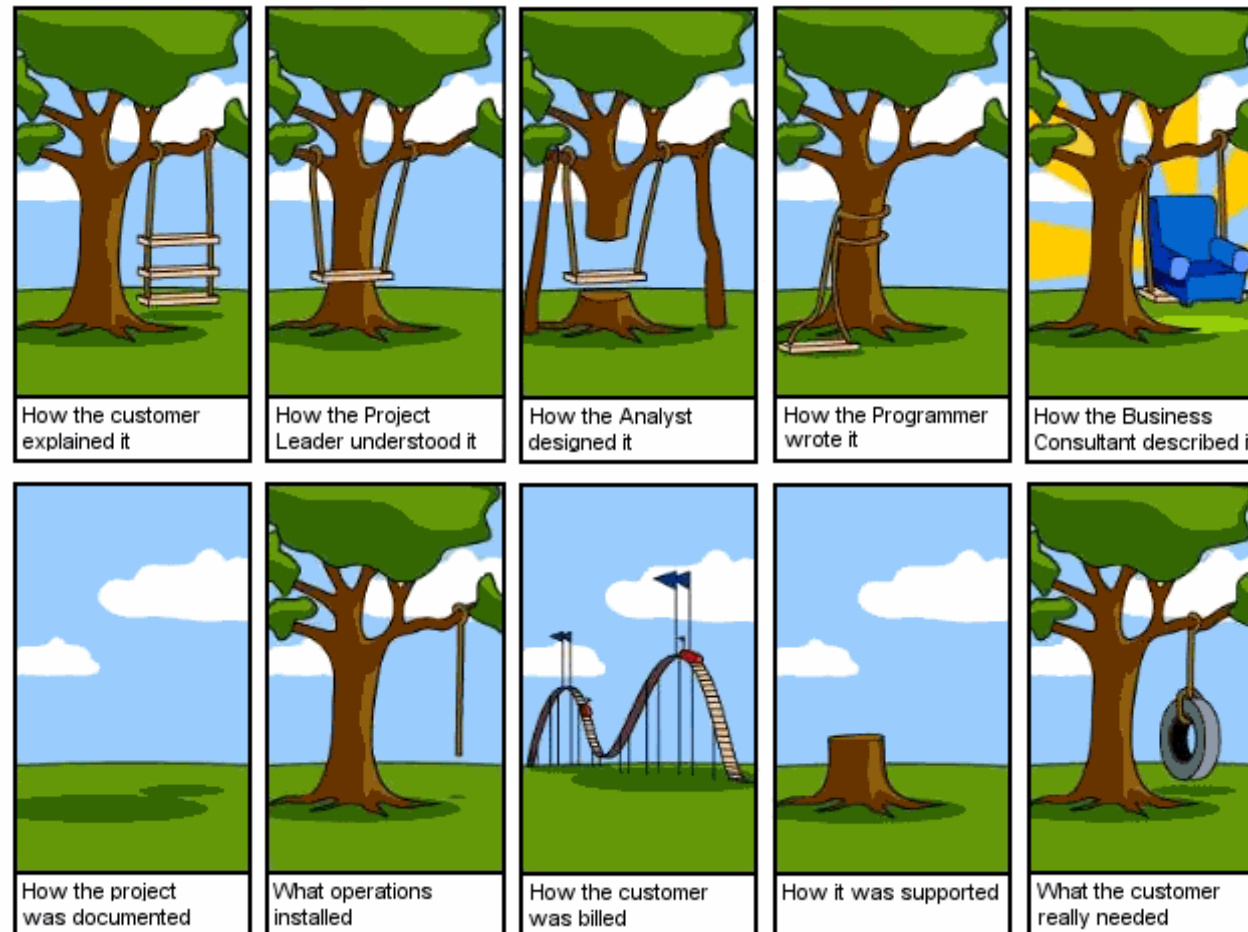
Recapitulare – probleme de integrare

- Probleme de natură tehnică
 - Conflicte de conexiune
 - Conflicte sintactice
 - Conflicte de control
 - Conflicte legate de calitatea serviciului
 - Conflicte legate de consistența datelor
 - Conflicte semantice
 - Conflicte funcționale



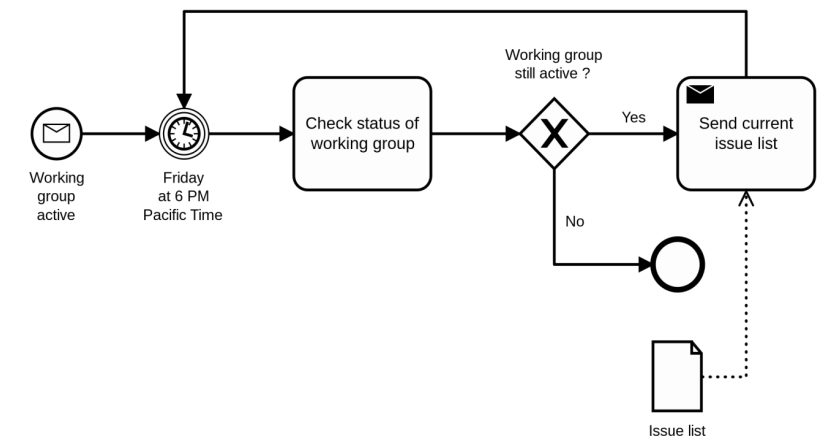
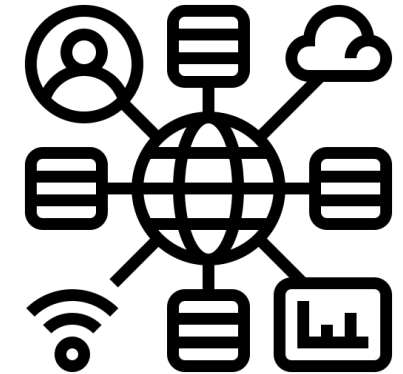
Image by Freepik

Recapitulare – probleme de integrare



Procesul de integrare

- Înțelegerea **domeniului** organizației și al problemelor
- Înțelegerea **datelor** – identificare, modelare
- Înțelegerea **proceselor** de afaceri – modelare, integrare
- Identificarea **interfețelor** aplicațiilor
- Identificarea **evenimentelor** din sistem
- Identificarea scenariilor de **transformare a datelor**
- Maparea **transferurilor de informație**
- Utilizarea **tehnologiilor**
- Evaluarea **rezultatelor** – testarea, performanța
- Crearea procedurilor de **întreținere**



Înțelegerea domeniului organizației

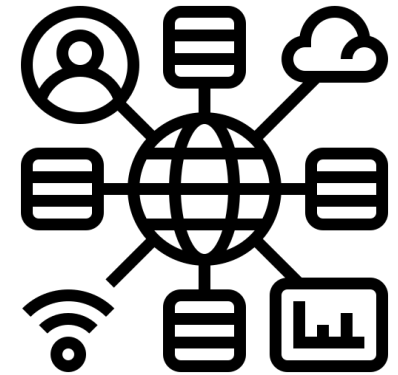
- Analiza proceselor de afaceri – aspect complex care ocupă cea mai mare durată a întregului proces
 - Este o problemă de identificare a cerințelor
- Necesită interfațarea cu
 - documente
 - persoane
 - sisteme
- în vederea
 - analizei
 - modelării
 - redefinirii cerințelor



Image by Freepik

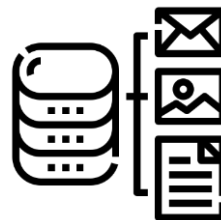
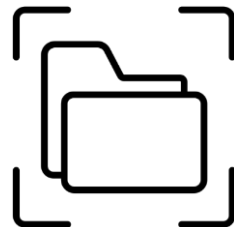
Înțelegerea datelor

- Cele mai multe proiecte de integrare se rezumă la **nivelul datelor**
- Indiferent de nivelul la care se realizează integrarea, este necesară înțelegerea datelor:
 - *unde* sunt datele
 - *ce* reflectă datele (structura informației)
 - *care* este fluxul datelor
 - *ce* transformări au loc și *de ce*



Înțelegerea datelor

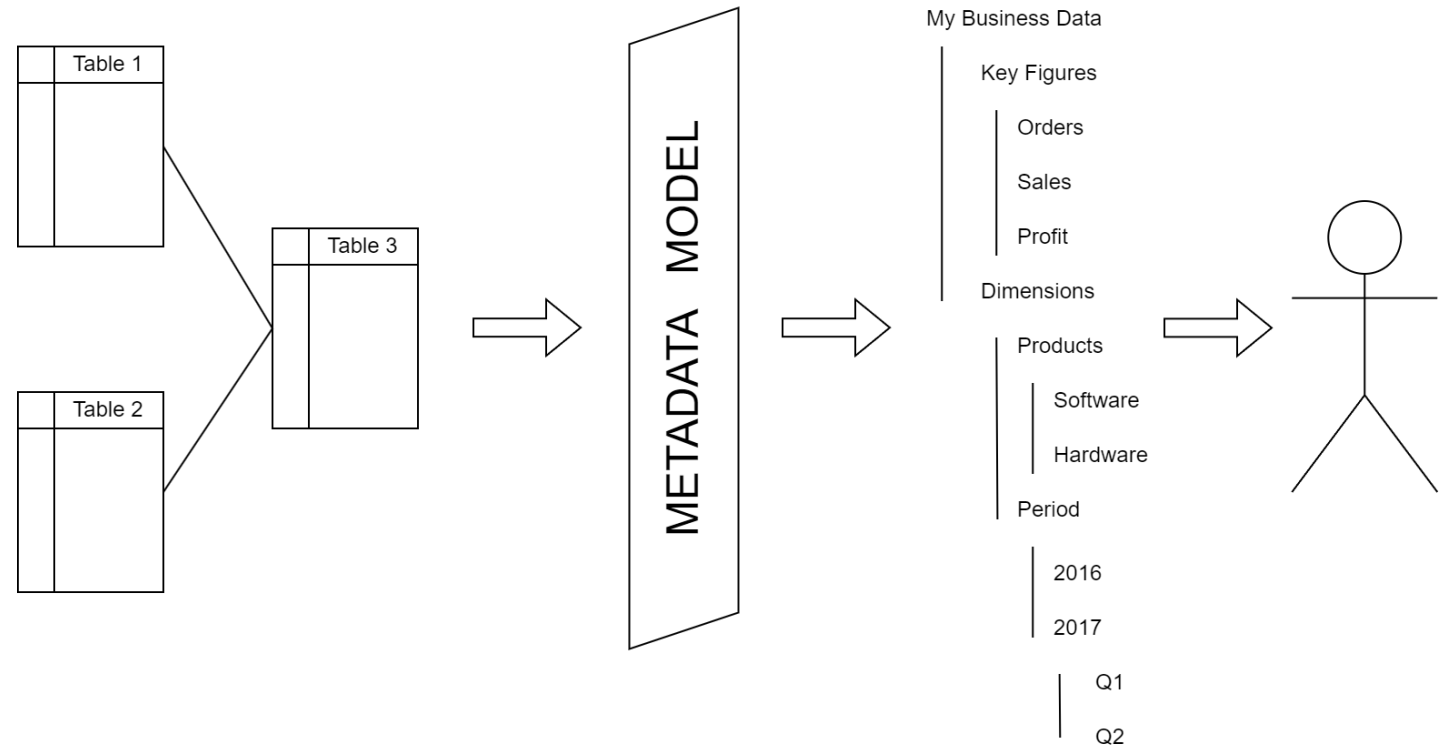
- Pentru integrarea la nivelul datelor sunt necesari 3 pași:
 - **Identificarea** datelor
 - **Catalogarea** datelor
 - Construirea modelului **metadatelor** organizației
 - *enterprise metadata model*



Înțelegerea datelor

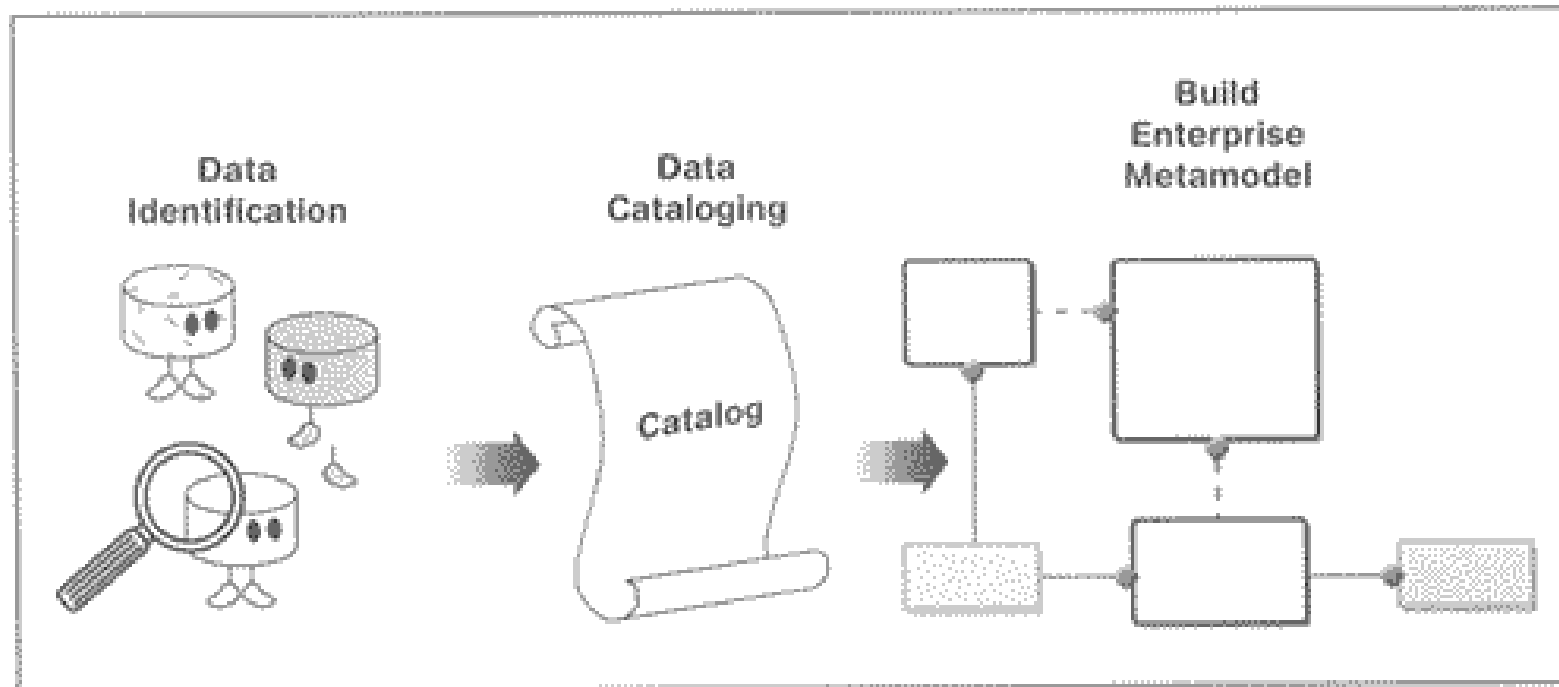
- Folosind metadata

- Titlu/nume
- Descriere
- Categorie
- Cuvinte cheie
- Data actualizării
- Informații de contact
- Licență
- ---
- Frecvență
- Acoperire temporală/spațială



Înțelegerea datelor

- O soluție de succes necesită definirea atât a modului în care este transferată informația, cât și a modului în care aceasta contribuie la derularea afacerii.

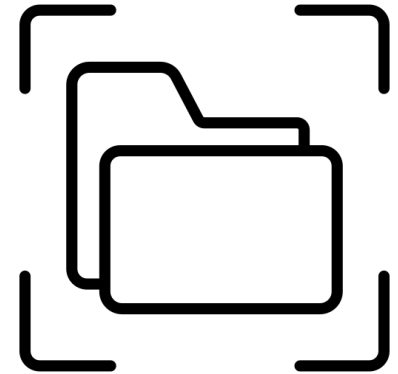


Înțelegerea datelor

> Identificarea datelor

De cele mai multe ori, informația – tehnică sau de business – este împrăștiată în organizație, în mod nestructurat, mai mult sau mai puțin utilizabilă

- Crearea unei liste a sistemelor sursă de date
- Identificarea
 - proprietarului datelor
 - localizării fizice a datelor
 - Informațiilor de proiectare relevante (model, tehnologia folosită, versiunea, etc.)



Înțelegerea datelor

Dicționarul datelor

- **Formatul** – cum sunt stocate datele
- **Latența** – când/cât de des sunt stocate datele
- **Integritatea** – accesibilitate, proprietate, parametrii de securitate, etc.

data/

OSF Storage (United States)

0.1 datadictionary.csv

0.2 raw.pdf

0.3 clean.csv

Tags

Sheet_1

Show rows with cells including:

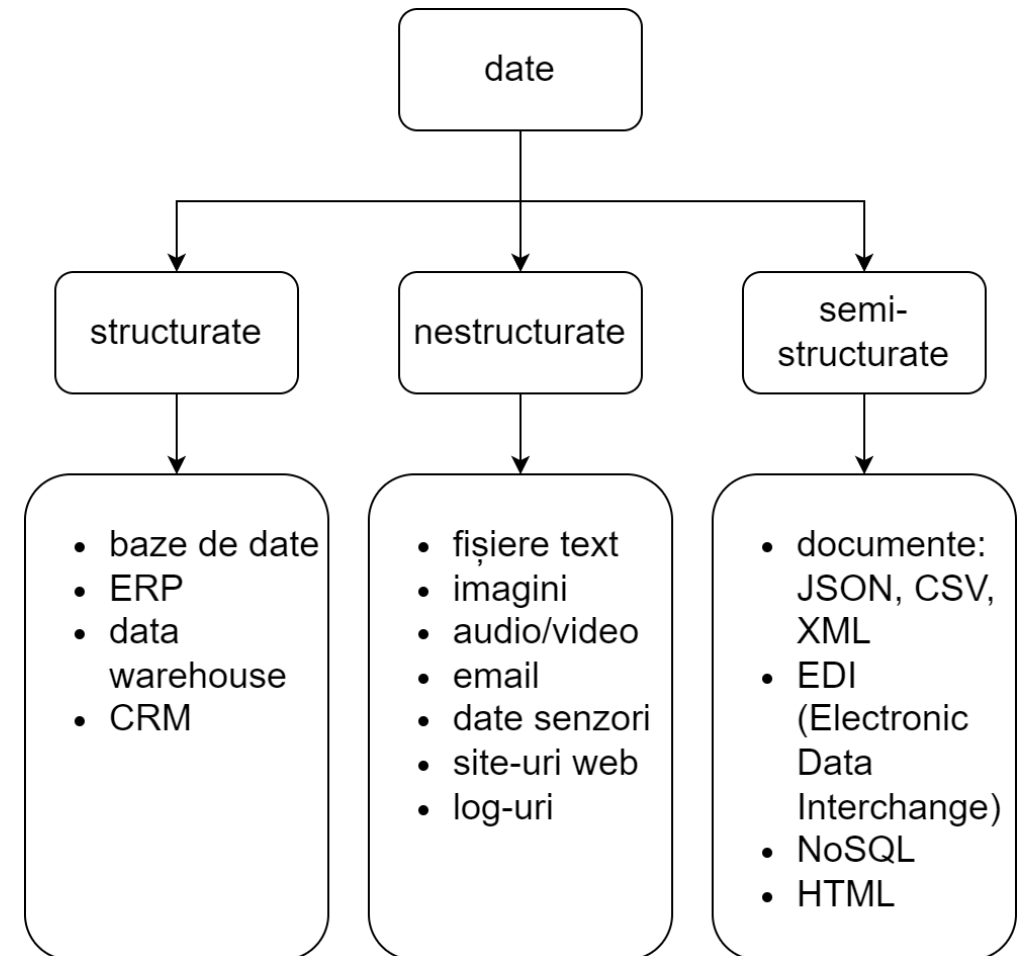
Variable	Variable name	Mesaurement unit	Allowed values	Description
Participant ID number	ID	Numeric	001-999	ID number assigned to participant in sequential order
Group number	GROUP	Numeric	1-30	Group assigned to participant based on ID number
Age in years	AGE	Numeric	18.0-65.0	Age of participant in years
Date of birth	DOB	mm/dd/yyyy	1-12/1-31/1951-1998	Participant's date of birth
Gender	SEX	Numeric	1 = male 2 = female	Participant's gender
Date of survey	SURVEY	mm/dd/yyyy	01/01/2015 – 01/01/2016	When the participant completed the survey
Self-reported consumer spending	SPEND	Numeric	0-100,000,000	Self-reported average yearly expenditure
Market sentiment	SENTIMENT	Numeric	1 = negative 2 = neutral 3 = positive	Sentiment towards US domestic economy
Actual GDP growth	GDP	Numeric	-5.0-5.0	Average US yearly GDP growth

Înțelegerea datelor

Formatul datelor

Cum este reprezentată informația

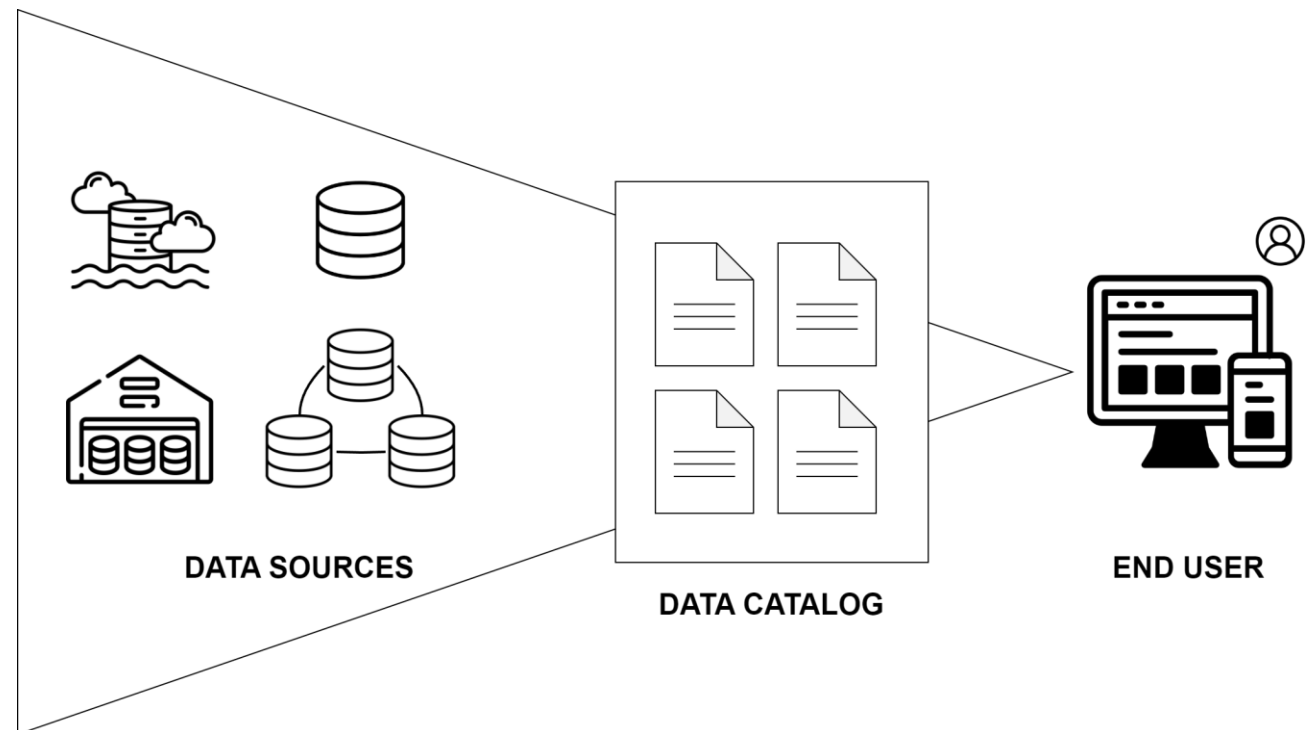
- numele elementului de date
- tipul datelor (caracter, numeric, logic, etc.)
- formatul de stocare (binar, text, etc.)



Înțelegerea datelor

> Catalogarea datelor

- este procesul de colectare de metadate din domeniul problemei
- Informații despre
 - sisteme
 - securitate
 - proprietari
 - procese conectate
 - mecanisme de comunicare
 - aspecte de integritate
 - etc.



Înțelegerea datelor

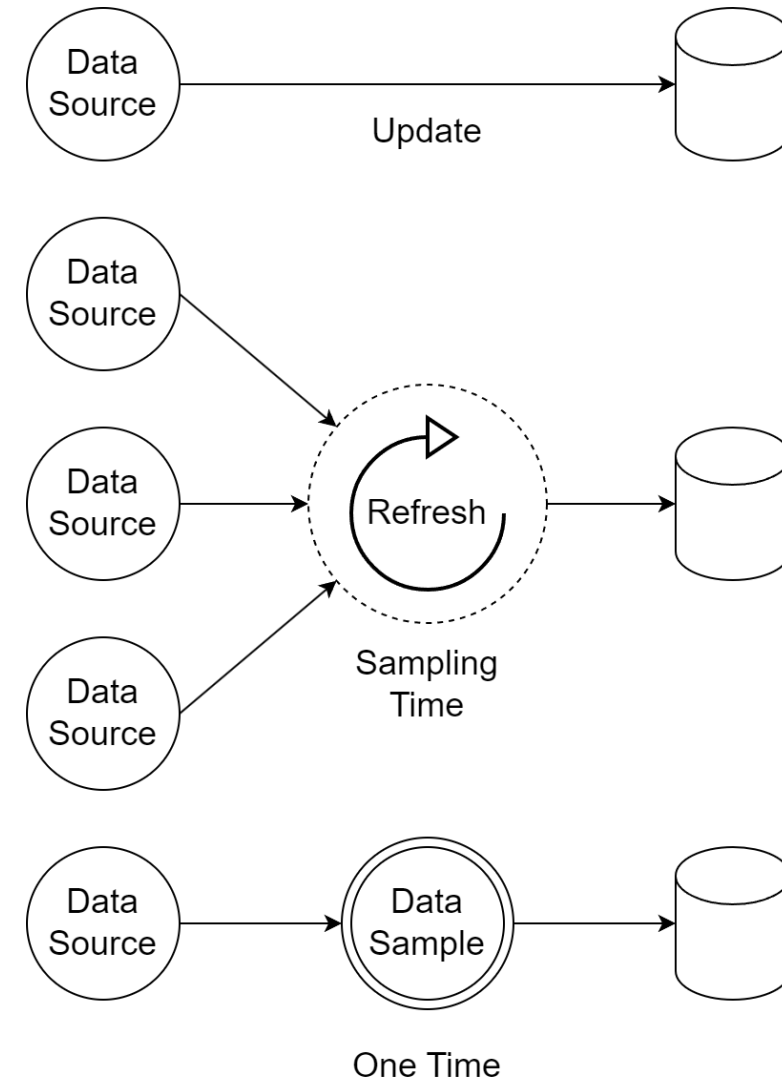
Latența datelor

- Arată cât de des trebuie actualizate datele
- Permite arhitecților de sisteme integrate să determine dacă și cât de des este necesară copierea/mutarea/replicarea informației
 - **Timp real** (en. real-time) – informația este plasată în baza de date imediat de este produsă, latența este foarte mică sau absentă
 - **Aproape timp real** (en. near-real-time) – datele se referă la informații care sunt actualizate la momente de timp prestabilite, nu neapărat instantaneu (datele sunt disponibile numai atunci când sunt necesare)
 - **Unice** (en. one-time) – datele sunt actualizate doar o dată

Înțelegerea datelor

Latența datelor

- **Timp real** (en. real-time) – informația este plasată în baza de date imediat de este produsă
- **Aproape timp real** (en. near-real-time) – datele sunt actualizate la momente de timp prestabilite
- **Unice** (en. one-time) – datele sunt actualizate doar o dată



Înțelegerea datelor

Aspecte de integritate

- Înțelegerea regulilor și reglementărilor aplicate la construcția bazei de date
- Lipsa controlului de integritate la nivelul datelor, sau ignorarea acestuia, poate duce la probleme majore
 - Desincronizări / inconsistențe la integrare
 - Probleme de trasabilitate (cine/ce/când/unde/de ce a actualizat datele)
 - Dependența de sisteme legacy
 - Probleme de mentenanță și securitate

Înțelegerea datelor

Aspecte de integritate

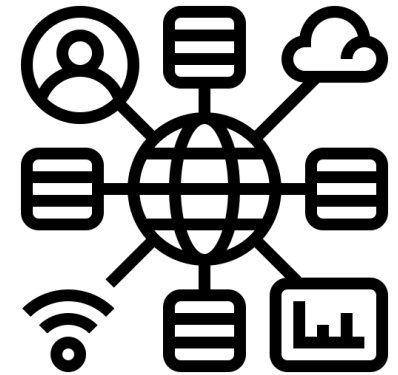
- Principiile ALCOA+ aplicate datelor
 - **Atribuibile** (autor identificabil)
 - **Lizibile** (accesibile)
 - **Contemporane** (înregistrate la momentul creării)
 - **Originale** (verificabile)
 - **Autentice** (documentate)
 - ---
 - **Complete** (întreg contextul)
 - **Consistente** (workflow-uri repetabile)
 - **Durabile** (accesibile mult timp)
 - **Disponibile** (accesibile oricând la nevoie)



Înțelegerea datelor

> Construirea modelului metadatelor organizației

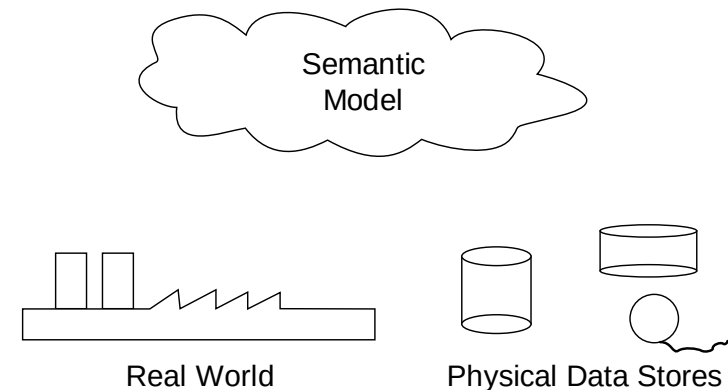
- Descrie **structura** datelor existente în organizație și modul în care acestea **interacționează** în soluția integrată
- Un depozit (repository) de metadata contribuie la **integrarea la nivelul datelor** și, în același timp, constituie **baza** pentru alte niveluri de integrare (ex. la nivel de metodă, de interfață utilizator, interfață de aplicații / API)



Construirea modelului datelor

Modelul fizic

- Nu există un mod simplu de a crea un model fizic care acoperă toate tipurile de baze de date ce pot exista într-o organizație (orientate obiect, multidimensionale, ierarhie, fișiere “flat” – text sau binare, baze de date relaționale, etc.)
- **Dar dacă aceste baze de date trebuie să fie integrate, este necesară o reprezentare fizică comună**
- Intrările modelului fizic sunt:
 - modelul logic
 - catalogul datelor



Construirea modelului datelor

Normalizarea organizației

- Prin normalizarea modelului se înțelege descompunerea structurilor de date complexe în relații simple, folosind o serie de reguli de dependență
- **Normalizarea înseamnă** reducerea volumului de date redundante la nivel de baze de date sau la nivel de organizație
- Se recomandă normalizarea modelului
 - la nivel logic
 - la nivelul fizic al proiectării bazelor de date
 - în cazul proiectului de integrare

Construirea modelului datelor

Exemplu: Normalizarea bazelor de date

- UNF – datele pot fi redundante și complexe
- 1NF – datele compuse / multiple sunt separate (coloane, rânduri)
 - atomicitatea datelor (indivizibile)
- 2NF – datele depind funcțional doar de cheia primară / dependențele parțiale sunt descompuse în tabele separate
 - prevenirea duplicării datelor
- 3NF – datele nu depind unele de altele / dependențele tranzitive sunt descompuse în tabele separate
 - prevenirea inconsistențelor

Construirea modelului datelor

Exemplu: normalizarea bazelor de date

Employee				
ID	name	managerID	sectorID	sectorName
1	David D.	1	4	Finance
2	George G.	1	3	IT
3	James J.	2	2	Security
4	Henry H.	2	1	Administration

2NF, 3NF

Sector	
ID	name
1	Administration
2	Security
3	IT
4	Finance

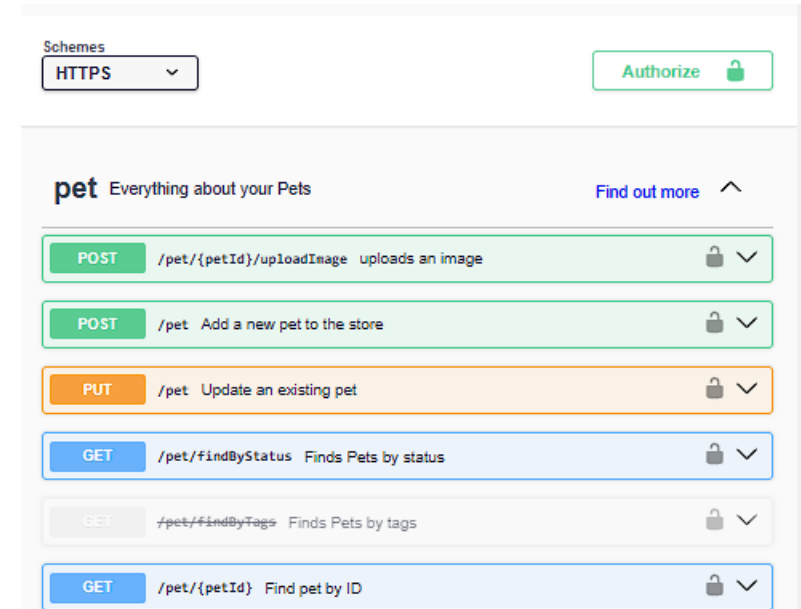
Manager			
ID	name	area	employees
1	Adam A.	East	David D. George G.
2	Betty B.	West	James J. Henry H.
3	Carl C.	North	

1NF

Înțelegerea proceselor

Procese de afaceri (business processes)

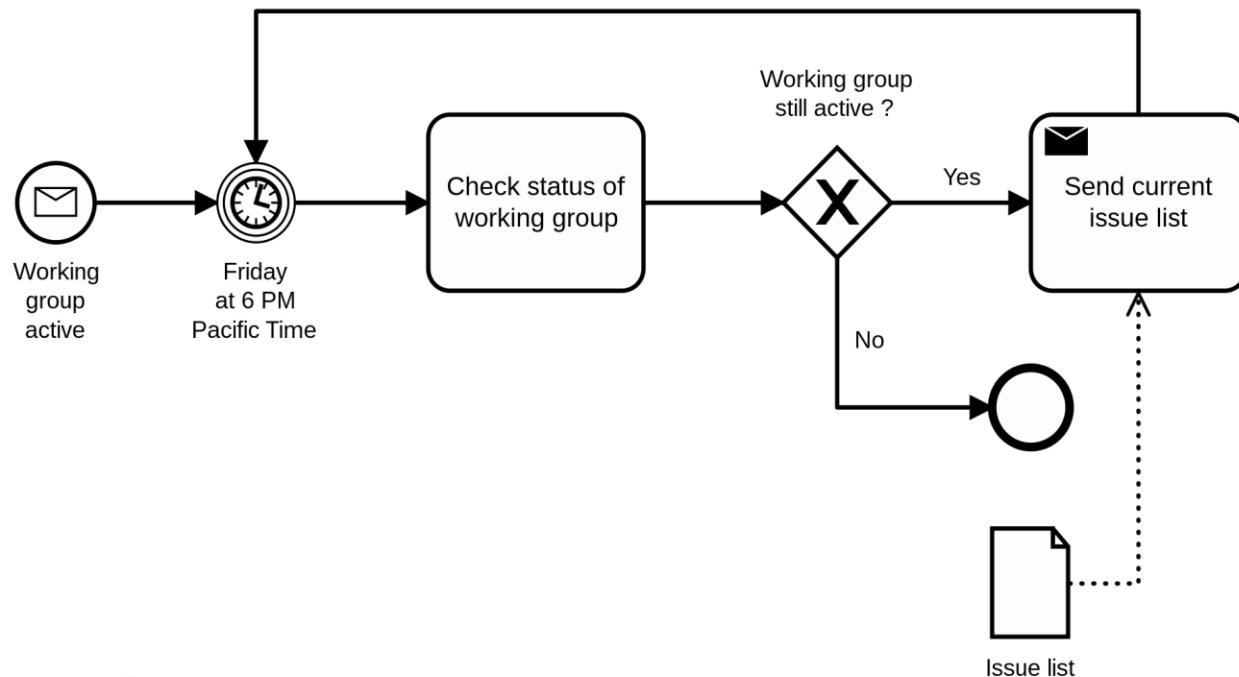
- sunt descrise prin **listarea funcțiilor** sau metodelor oferite de o aplicație (function / method listings) /sau a interfețelor expuse
- În anumite aplicații, sunt ușor de identificat deoarece procesele de afaceri sunt **bine documentate** sau intuitive (ușor de găsit prin invocarea interfeței utilizator)
- În alte aplicații, determinarea acestor procese necesită căutarea la nivelul codului sursă pentru a înțelege diferitele funcții și metode disponibile



Ex. OpenAPI (Swagger)

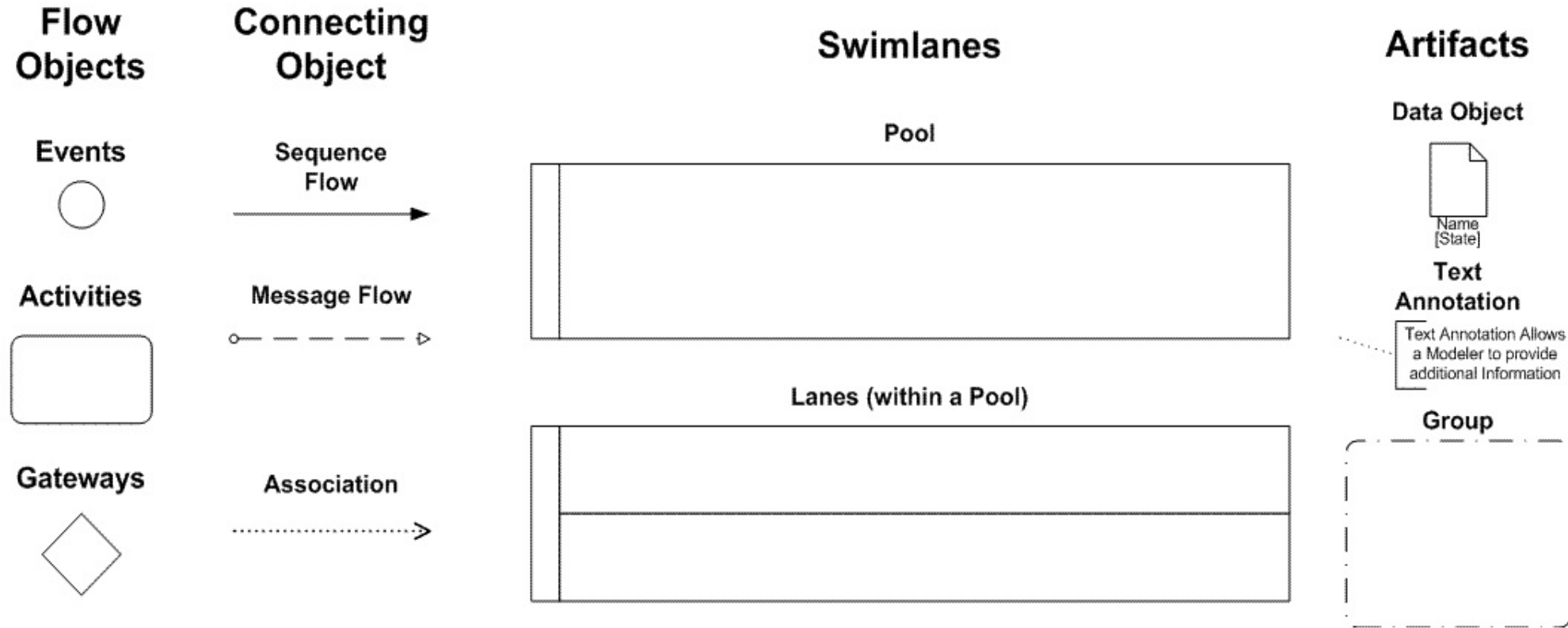
Modelarea proceselor – BPMN

- BPMN (Business Process Model and Notation) este un limbaj grafic de definire a proceselor/workflows
 - Reprezentare grafică
 - Mapare pe cerințele de business
 - Similar diagramei de activități (UML)



Sursa: Mikelo Skarabo - BPMN - A Process with normal flow

Modelarea proceselor – BPMN



Integrarea proceselor

- Trebuie luată decizia privind modul de abordare al proceselor de afaceri în implementare
- Se dezvoltă o perspectivă la nivelul **metodelor** și proceselor
- Se documentează toate **procesele de afaceri** și modul în care acestea relaționează între ele
- Se documentează **modelul metadatelor** întreprinderii
- Se recomandă folosirea tehnicilor de modelare tradiționale (**UML**) pentru a crea procesele de afaceri
- Pentru procesele de afaceri existente se va analiza modalitatea de **integrare la nivelul metodelor**, printr-o aplicație compozită

Integrarea proceselor

Integrarea proceselor

- Necesită **analiza tuturor aplicațiilor** care există/sunt folosite în organizație – pot fi foarte diferite ca tip
- Este necesara **stabilirea tehnologiilor folosite** pentru fiecare aplicație în parte – este posibil să coexiste aplicații bazate pe tehnologii vechi de 30 de ani cu aplicații de ultimă generație
- Trebuie stabilit **proprietarul procesului**, apoi trebuie înțeles modul de procesare adoptat
- Trebuie identificate **documentațiile** tuturor proceselor – o documentație corespunzătoare are un impact major asupra procesului de integrare

Integrarea proceselor

Integrarea **la nivelul metodelor** necesita întocmirea unui “catalog al proceselor” – o listă a tuturor proceselor de afaceri din domeniul afacerii

- scopul derulării procesului
- cine e proprietarul
- ce face exact
- ce tehnologie utilizează
- cu ce baze de date interacționează
- cum se leagă de modelul metadatelor
- descriere sub formă de pseudocod /diagrama fluxului de date /diagrama de stare pentru a descrie fluxul controlului între procese

Integrarea proceselor

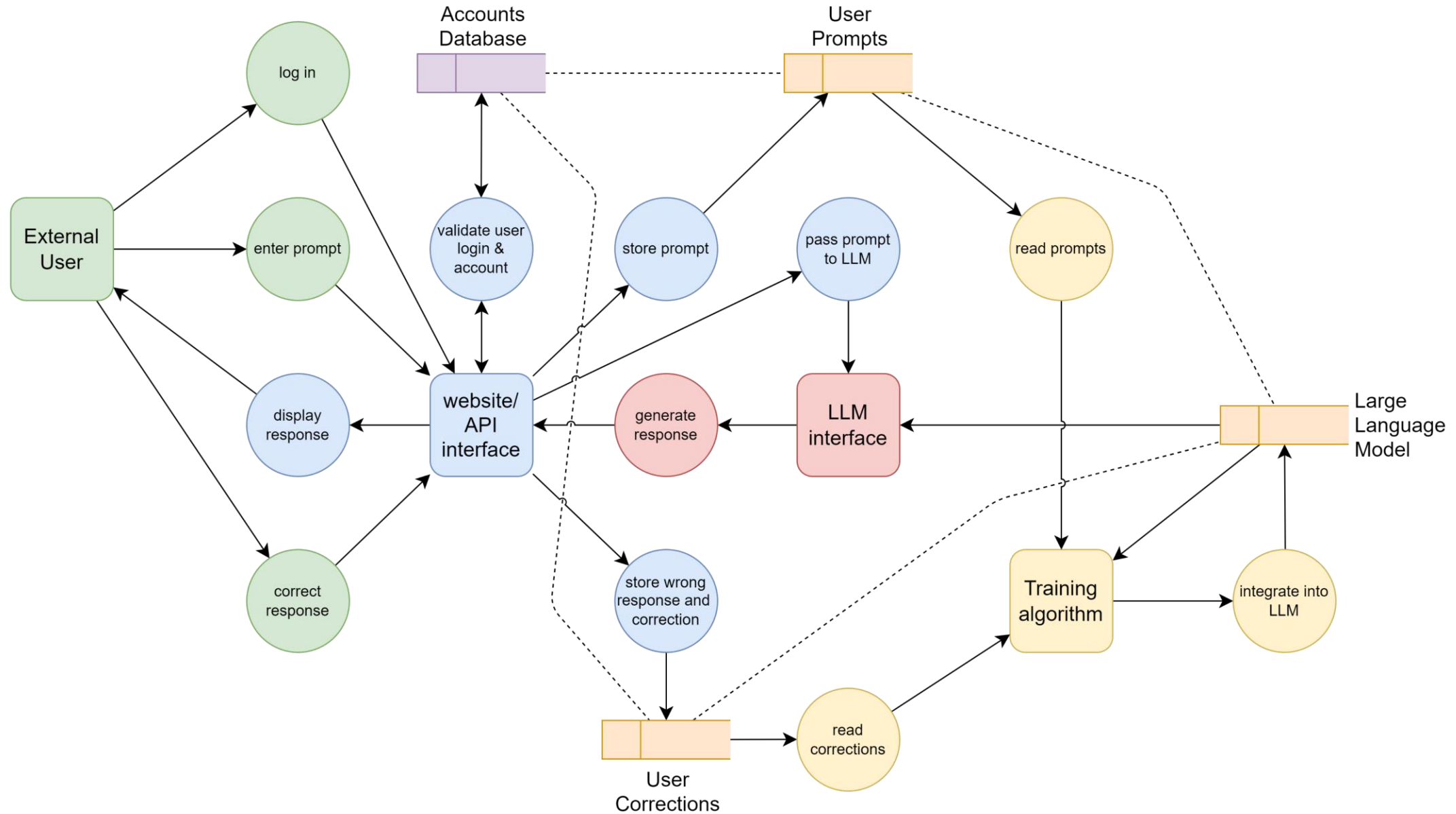
Modelul de afaceri

- Este rezultatul agregării, într-un model de nivel înalt, a tuturor obiectelor, metodelor, proprietăților, procedurilor și procesărilor, sau a altor transformări la care este supusă informația în organizație = **stadiul final al integrării la nivel de metodă**
- NU redă un model complex, orientat pe detalii
- Este axat pe **înțelegerea proceselor** în contextul organizației, a funcțiunilor și caracteristicilor acestora, precum și a posibilităților de combinare și reutilizare a proceselor

Integrarea proceselor

- Scopul final al integrării este de a depăși simpla conexiune între aplicații pentru a ajunge la definirea și proiectarea unui **flux al proceselor de afaceri**
- Pentru a realiza acest lucru, trebuie documentate atât starea actuală și starea dorită, cât și pașii ce trebuie urmați

Modelarea proceselor



Integrarea proceselor

- Modelul trebuie sa rămână o paradigmă **independentă de implementare**
 - Multe dintre aplicațiile dezvoltate în ultimii zece ani folosesc paradigma **orientată-obiect** atât pentru proiectare cât și pentru programare
 - Cu toate acestea, cele mai multe aplicații tradiționale folosesc paradigma de programare și proiectare **structurată** (module, funcții)
 - Modelul de afaceri trebuie să rămână **independent de ambele**

Modelarea proceselor

Folosirea **modelelor** presupune găsirea unei **structuri comune** care să rezolve problemele implicate în realizarea proceselor

- **Modele arhitecturale (architectural patterns)**
 - Oferă arhitecților **structuri de bază** ale sistemului/subsistemului, modele de comportament și relații
- **Modele de proiectare (design patterns)**
 - Oferă arhitecților/dezvoltatorilor **mijloace pentru definirea subsistemelor** și componentelor, precum și a relațiilor dintre ele

Identificarea interfețelor aplicațiilor

Trebuie identificate **interfețele** disponibile ale diferitelor aplicații, în vederea integrării **la nivelul interfețelor aplicațiilor / API**

- Este necesară crearea unui **director al interfețelor aplicațiilor** = un repository care conține informațiile disponibile despre interfețe, precum și documentația aferentă (ex. [OpenAPI](#))
- Este esențială înțelegerea tuturor proceselor de afaceri care sunt disponibile/susținute prin intermediul unei aplicații
- Este posibil să fie necesară identificarea unor combinații particulare de procese ce trebuie invocate pentru a realiza o anumită cerință de integrare (workflow)

Identificarea interfețelor aplicațiilor

Directorul interfețelor aplicației (application interface directory)

- Poate fi gândit ca o listă a proceselor de afaceri disponibile într-o aplicație (packaged sau custom-made)

Semantica aplicației

- Descrie modul și forma în care o aplicație particulară adresează proprietățile procesului de afaceri
- Ce înseamnă un anume termen într-o anume aplicație

The screenshot displays the Swagger Editor web application. The left pane shows the OpenAPI 3.0 definition for the Swagger Petstore API, including details like title, description, contact, license, and endpoints. The right pane shows the rendered API documentation, featuring the title 'Swagger Petstore - OpenAPI 3.0', a description, useful links, terms of service, and a list of API endpoints with their methods and descriptions.

```
1 openapi: 3.0.3
2 info:
3   title: Swagger Petstore - OpenAPI 3.0
4   description: |-
5     This is a sample Pet Store Server based on the OpenAPI 3.0 specification. You can find out more about
6     Swagger at [https://swagger.io](https://swagger.io). In the third iteration of the pet store, we've switched
7     to the design first approach!
8     You can now help us improve the API whether it's by making changes to the definition itself or to the code.
9     That way, with time, we can improve the API in general, and expose some of the new features in OAS3.
10
11   _If you're looking for the Swagger 2.0/OAS 2.0 version of Petstore, then click [here](https://editor.swagger
12   .io?url=https://petstore.swagger.io/v2/swagger.yaml). Alternatively, you can load via the `Edit > Load
13   Petstore OAS 2.0` menu option!_
14
15   Some useful links:
16   - [The Pet Store repository](https://github.com/swagger-api/swagger-petstore)
17   - [The source API definition for the Pet Store](https://github.com/swagger-api/swagger-petstore/blob/master
18   /src/main/resources/openapi.yaml)
19
20 termsOfService: http://swagger.io/terms/
21 contact:
22   email: apiteam@swagger.io
23 license:
24   name: Apache 2.0
25   url: http://www.apache.org/licenses/LICENSE-2.0.html
26 version: 1.0.11
27 externalDocs:
28   description: Find out more about Swagger
29   url: http://swagger.io
30 servers:
31   - url: https://petstore3.swagger.io/api/v3
32 tags:
33   - name: pet
34     description: Everything about your Pets
35     externalDocs:
36       description: Find out more
37       url: http://swagger.io
38   - name: store
39     description: Access to Petstore orders
40     externalDocs:
41       description: Find out more about our store
42       url: http://swagger.io
43   - name: user
44     description: Operations about user
45 paths:
46   /pet:
47     put:
48       tags:
49       - pet
50       summary: Update an existing pet
51       description: Update an existing pet by Id
52       operationId: updatePet
53       requestBody:
54         description: Update an existent pet in the store
55         content:
```

Swagger Petstore - OpenAPI 3.0 1.0.11 OAS 3.0

This is a sample Pet Store Server based on the OpenAPI 3.0 specification. You can find out more about Swagger at <https://swagger.io>. In the third iteration of the pet store, we've switched to the design first approach! You can now help us improve the API whether it's by making changes to the definition itself or to the code. That way, with time, we can improve the API in general, and expose some of the new features in OAS3.

If you're looking for the Swagger 2.0/OAS 2.0 version of Petstore, then click [here](#). Alternatively, you can load via the [Edit > Load Petstore OAS 2.0](#) menu option!

Some useful links:

- [The Pet Store repository](#)
- [The source API definition for the Pet Store](#)

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

Servers

<https://petstore3.swagger.io/api/v3> [Authorize](#)

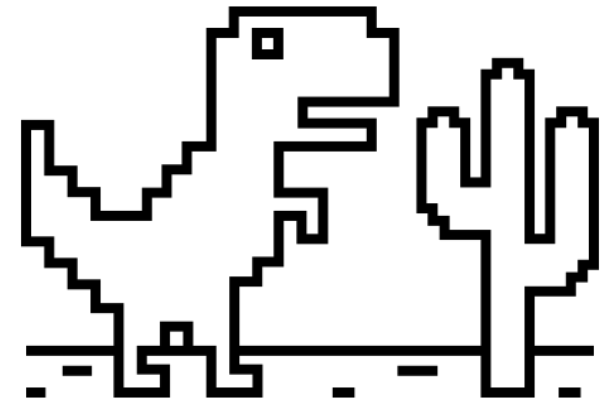
pet Everything about your Pets [Find out more](#)

PUT	/pet	Update an existing pet	Find out more
POST	/pet	Add a new pet to the store	Find out more
GET	/pet/findByStatus	Finds Pets by status	Find out more
GET	/pet/findByTags	Finds Pets by tags	Find out more



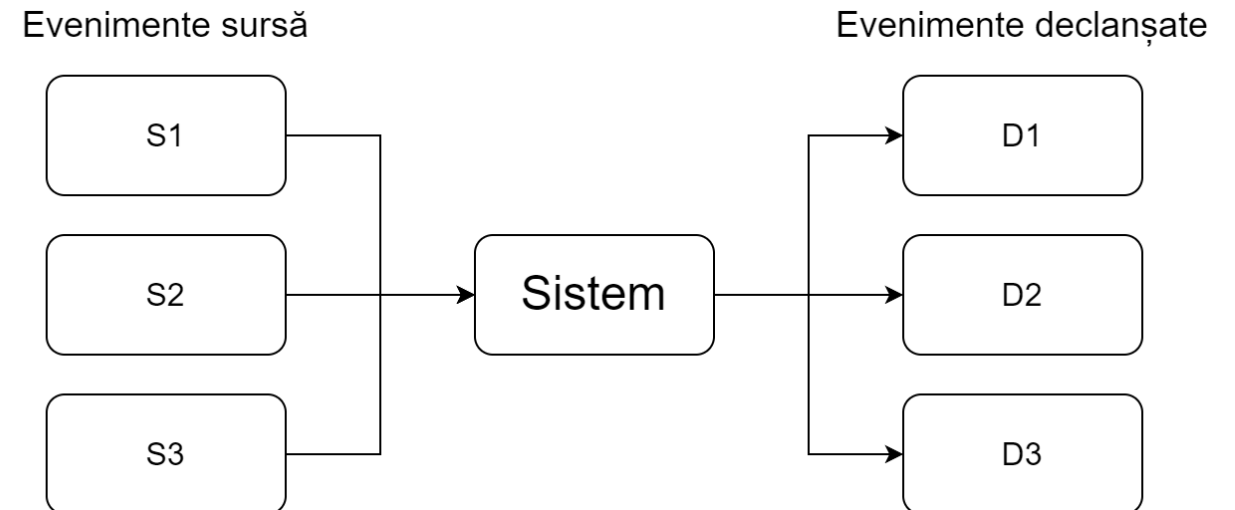
Etape adiționale

- Identificarea **evenimentelor** din sistem
- Identificarea **scenariilor** de transformare a datelor
- Maparea **transferurilor** de informație
- Utilizarea tehnologiilor
- Testarea
- Performanța
- Evaluarea rezultatelor
- Definirea procedurilor de întreținere



Identificarea evenimentelor din sistem

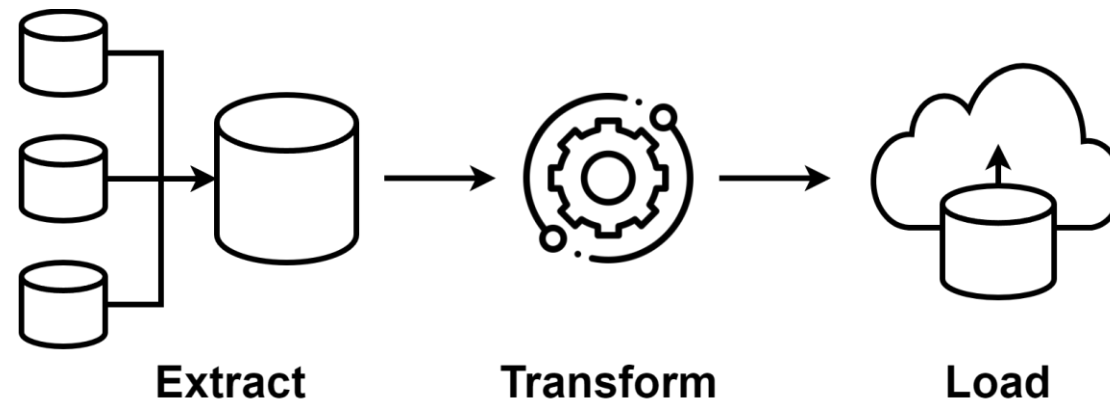
- Identificarea tuturor evenimentelor relevante pentru derularea proceselor
- Evenimentele sunt, de cele mai multe ori asincrone, dar pot avea instanțe sincrone
- Trebuie identificat / înțeles:
 - Ce declanșează un eveniment
 - Ce se întâmplă în timpul evenimentului
 - Ce alte evenimente sunt declanșate ca o consecință a evenimentului inițial



Identificarea scenariilor de transformare a datelor

Identificarea transformărilor la care sunt supuse structura și conținutul datelor, deoarece:

- datele dintr-un sistem ar putea să nu fie “înțelese” de sistemul destinație
- Trebuie asigurată **consistența semantică** a aplicației, de la sistem la sistem



Maparea transferurilor de informație

- Trasarea fluxului de informație între sisteme
 - de unde provine elementul de informație (data) și prin intermediul cărei interfețe
 - care este destinația informației
- Identificarea evenimentului care declanșează transferul informației sau
- Identificarea condiției care duce la transferul informației de la sursă la destinație (timp, variații ale unor parametrii, etc.)

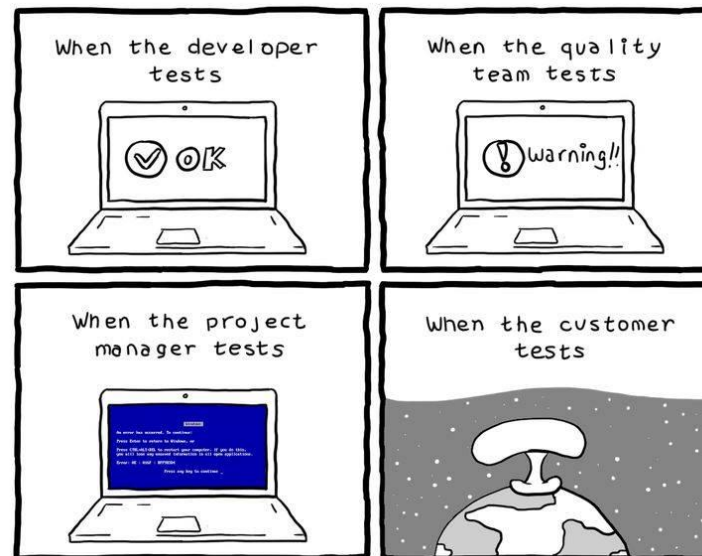
Utilizarea tehnologiilor

- Selectarea tehnologiei adecvate pentru a soluționa problema de integrare (ex. servere de aplicație, obiecte distribuite, broker de mesaje)
- De regulă, soluția necesită un mix de tehnologii
 - Condiția este ca ele să fie compatibile
- Definirea criteriilor pentru produse și tehnologii

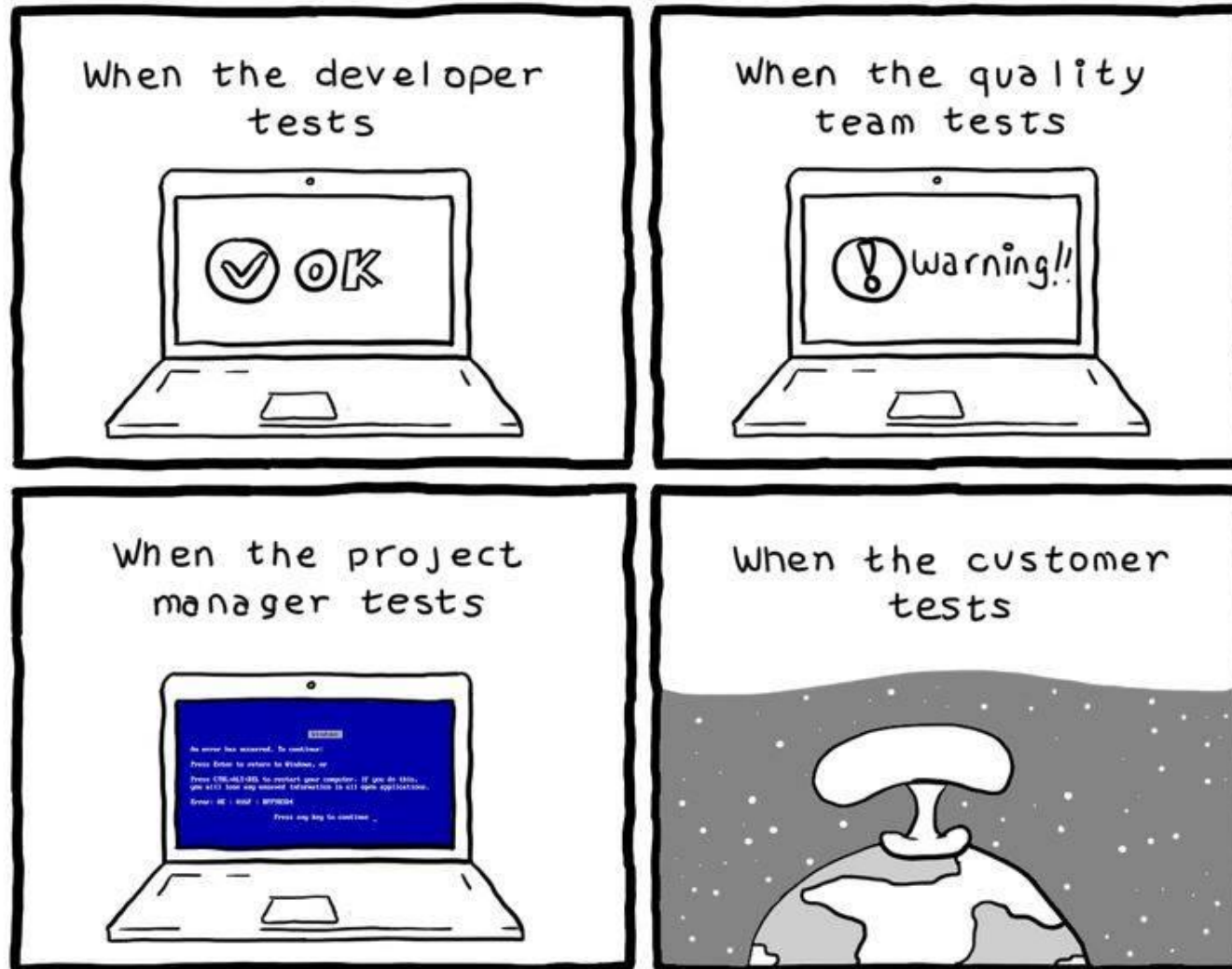


Testarea

- Testarea este costisitoare și necesită mult timp
- Trebuie să existe un plan de testare
- Testarea soluțiilor integrate este dificilă deoarece sistemele sursă și destinație sunt, de cele mai multe ori, critice și nu pot fi testate offline



Testarea



Performanța

- Performanța trebuie proiectată încă de la început. Arhitectura soluției integrate trebuie să ofere infrastructura și tehnologia necesare pentru performanța dorită
- De foarte multe ori, performanța nu este o prioritate decât în momentul când e prea târziu
- Performanța trebuie testată înainte de a da sistemul în folosință



Evaluarea rezultatelor

Orice soluție integrată trebuie să aducă valoare măsurabilă afacerii

- costul proceselor de afaceri eliminate prin folosirea soluției integrate (ex. automatizarea unor procese, reducerea erorilor, scurtarea duratei de procesare – hard \$ savings)
- creșterea în timp a productivității, creșterea satisfacției clienților, extinderea sistemului prin interoperabilitatea cu alte sisteme, etc. – soft \$ savings



Definirea procedurilor de întreținere

- Trebuie avut în vedere modul în care soluția integrată se poate întreține în timp. Ex:
 - Cine va administra serverul broker de mesaje?
 - Cine va gestiona aspectele de securitate?
 - Cine va monitoriza performanța sistemului și va soluționa problemele?
- Soluția de integrare este responsabilă de transferul informației între (sub)sistemele (critice) ale organizației, ceea ce înseamnă că rezultă și riscuri asociate
- Toate activitățile de întreținere trebuie documentate
- Problemele trebuie asignate direct unor responsabili



Întrebări?

