



Managementul proiectelor informatice



Constanta-Nicoleta Bodea
bodeaconstanta@gmail.com

Structura tematica

- I. Provocari si solutii in managementul proiectelor IT
- II. Notiuni de baza in managementul proiectelor: proiect, managementul proiectelor, program si portofoliu de proiecte
- III. Abordari agile in proiecte IT – SCRUM
- IV. Studiu de caz: Desfasurarea activitatilor intr-un proiect IT implementat in SCRUM
- V. Tehnici agile in proiecte IT
- VI. Instrumente software pentru implementarea agila a proiectelor IT



Cursul nu acopera:

- ▶ Aspectele tehnice ale domeniului IT (arhitecturi, metode, tehnici si instrumente pentru dezvoltarea/implementarea de sisteme software)
- ▶ Prezentarea de pachete software pentru managementul proiectelor, chiar daca se vor utiliza instrumente software pentru managementul agil al proiectelor.



Finalizarea disciplinei

- ▶ **Proiect (5 puncte)**
- ▶ **Examen (4 puncte)**
- ▶ **Punctaj din oficiu (1 punct)**

Proiectul de disciplina va fi predat la examen, in format electronic si va cuprinde:

- documentatia de proiect (.doc sau .pdf)
- date cont de conectare la aplicatie (se va defini un membru de echipa cu numele cbodea, si adresa de e-mail bodeaconstanta@gmail.com)

Proiectul poate fi elaborat individual sau in echipe de maxim 3 masteranzi.

Cerinte pentru proiectul de disciplina

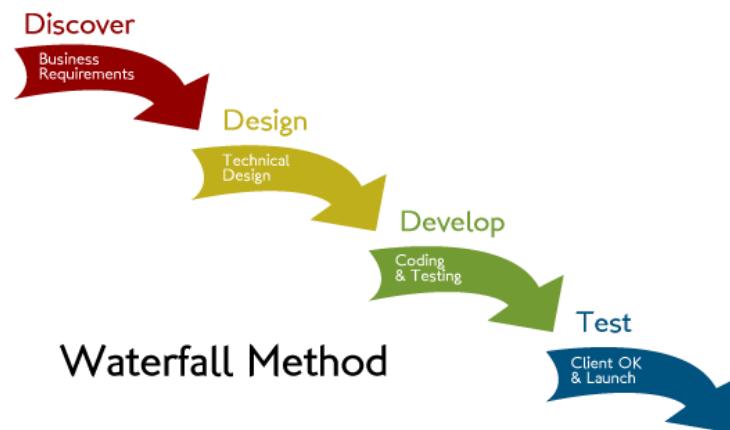
- ▶ Proiectul trebuie sa detalieze modalitatea în care este implementată metodologia SCRUM intr-un proiect de dezvoltare software, pentru un produs software ales de către echipa de masteranzi/masterand. Ca instrument de lucru se va utiliza *Jira*.

PARTEA I

Proiectele IT – provocari si solutii

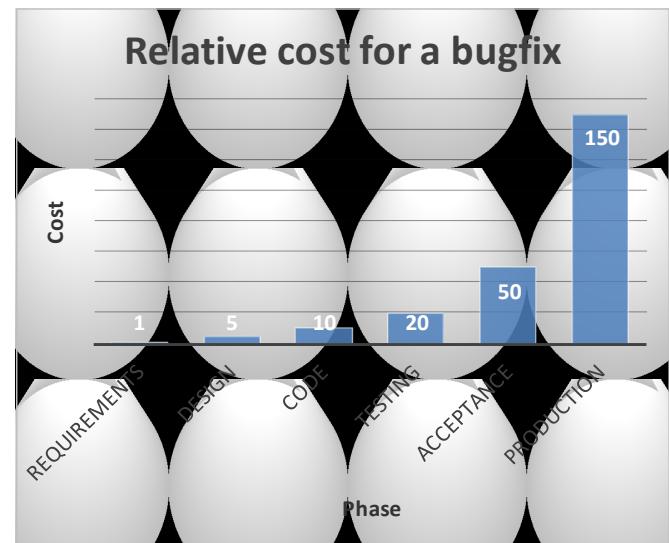
Provocari in implementarea proiectelor IT

- ▶ Waterfall reprezinta o abordare frecventa de dezvoltare secventiala;
- ▶ Urmareste secvential 7 faze:
 - ▶ Requirements specification (Product requirements document)
 - ▶ Design (Software architecture);
 - ▶ Construction/Implementation/Coding (actual software);
 - ▶ Integrare;
 - ▶ Testare;
 - ▶ Instalare;
 - ▶ Mantinanta;



Limite ale abordarii waterfall

- ▶ Este un model liniar si secvential si presupune ca faza precedenta sa se fi incheiat pentru a incepe o alta faza;
- ▶ Nu tine cont de faptul ca cerintele (requirement-urile) se schimba constant;
- ▶ Clientul nu are acces la produs pana cand acesta nu este complet;
- ▶ In functie de momentul (faza de dezvoltare) in care o eroare este descoperita, costul pentru a o repară creste exponentiial;
- ▶ Nu este un model potrivit pentru proiecte cu o durata mare de viata;



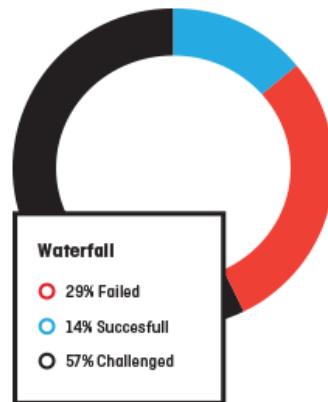
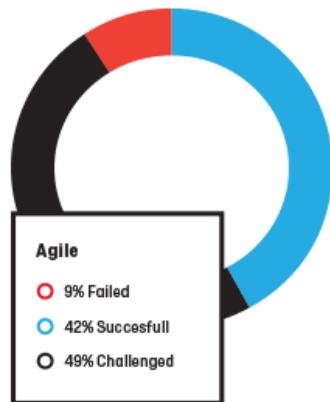
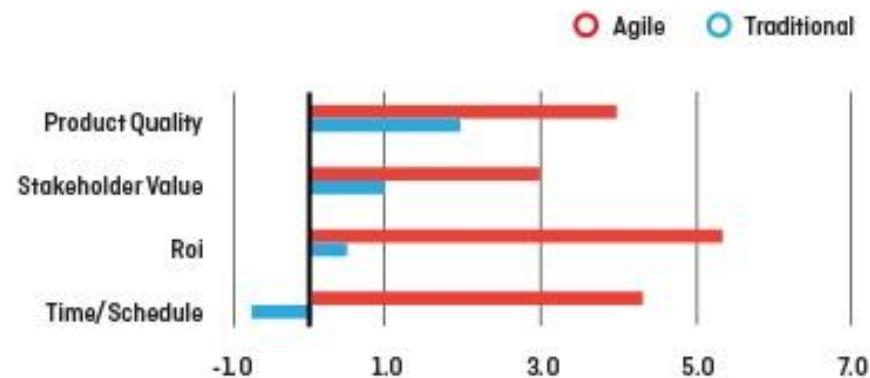
Rata de succes a proiectelor IT

Comparing IT project success rate by paradigm: 2013

- Agile
 - 8% Failed
 - 64% Succesfull
 - 28% Challenged

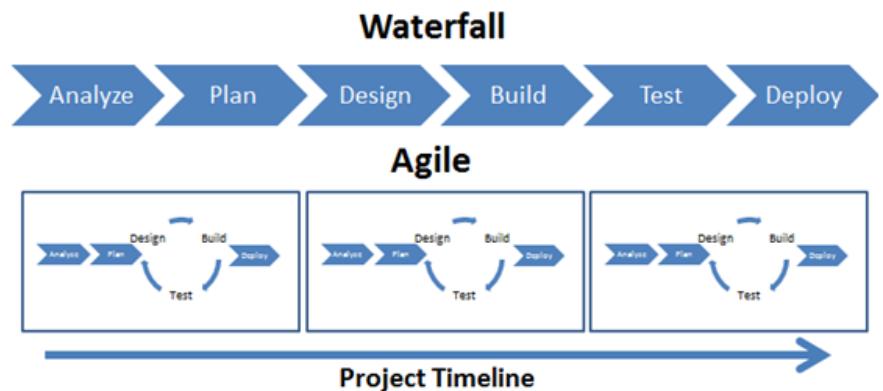
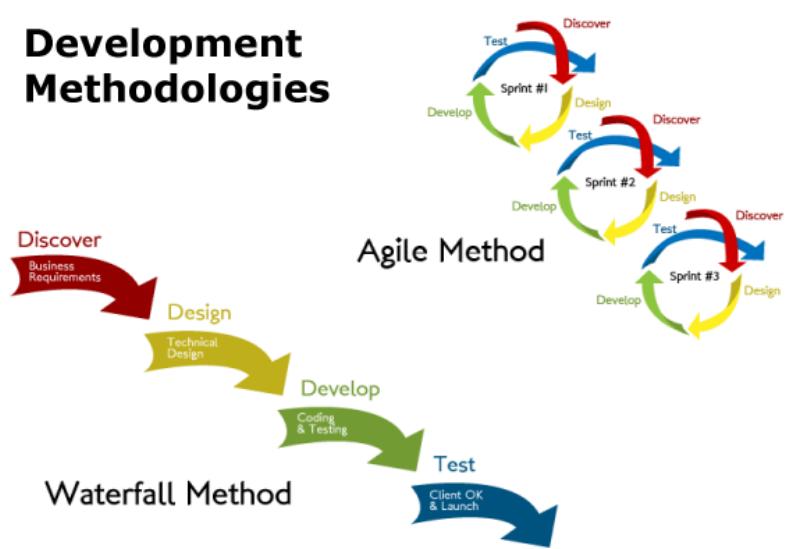
- Waterfall
 - 18% Failed
 - 49% Succesfull
 - 33% Challenged

Comparing software development paradigms (scale -10 to +10)



Waterfall vs. Agile

Development Methodologies



Alte provocarile care impun agilitatea

- ▶ Organizațiile trebuie să absoarbă și să se adapteze la **schimbari** mult mai rapide și profunde decât înainte. Cele mai bune practici își pierd din semnificativitate/utilitate.
- ▶ **Complexitatea** și **incertitudinea** sunt mai ridicate, impunând o altă formă de leadership (shared leadership).
- ▶ Concentrarea pe ceea ce este valoros pentru **clienți**.

- ▶ Agilitatea nu este un obiectiv *în sine*, ci ca un mijloc de asigurare a succesului proiectelor



Avantajele Agile

Waterfall – Long, Large, Linear, Late



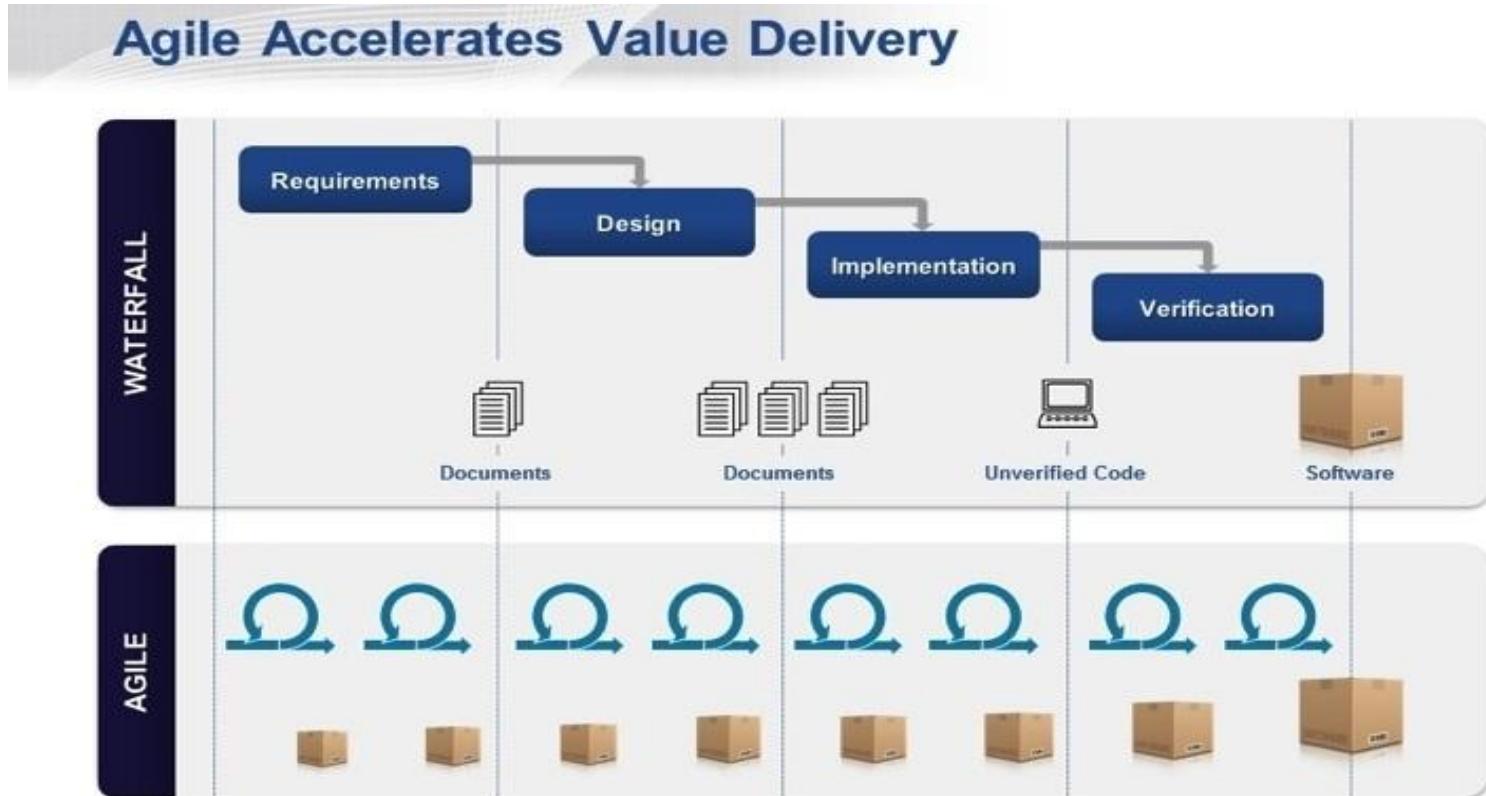
Agile: Iterate, Increment and Innovate



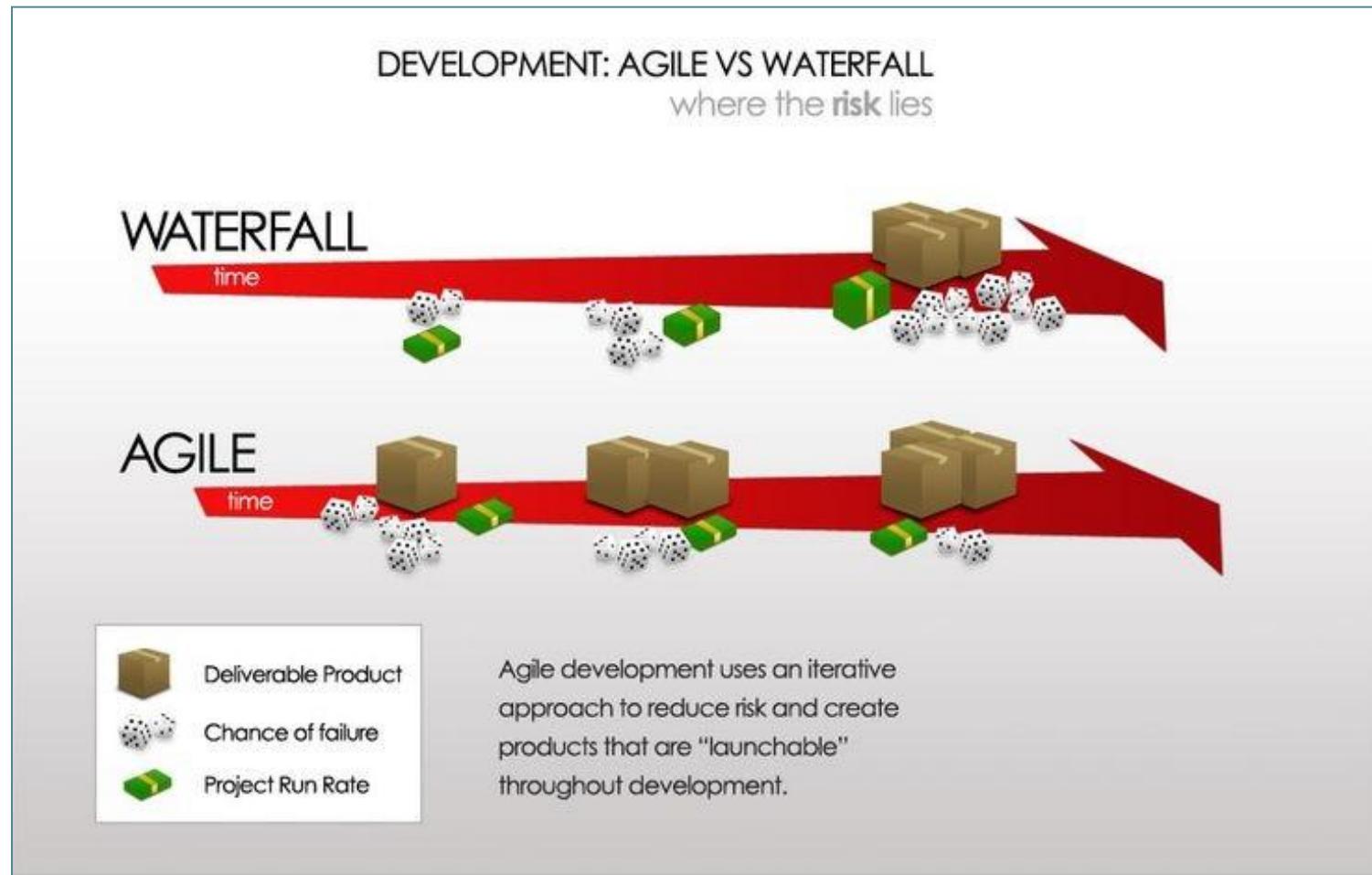
- ▶ Time to market;
- ▶ Flexibilitate sau agilitate;
- ▶ Calitate;
- ▶ Livreaza plus valoare;
- ▶ Minimizeaza riscurile;
- ▶ Creste eficienta si productivitatea;
- ▶ Permite o planificare mai buna pe termen lung;



Agile produce si accelereaza livrarea valorii



Aspecte legate de risc



Intrebări

1. Abordarea waterfall presupune parcurgerea secventială a urmatorilor pasi:

Faza	Numar de ordine
Specificare cerinte	
Dezvoltare cod	
Mentenanta	
Design	
Testare	
Instalare	
Integrare	

2. Principala limită a abordării waterfall este: _____



-
3. Enumerati trei avantaje ale abordarilor agile:

4. Explicati afirmatia: *Agile produce si accelereaza livrarea valorii.*
5. Cum se manifesta riscurile in proiectele IT implementate in abordare waterfall? Dar in cele cu implementare agila?



PARTEA II

Notiuni de baza in managementul proiectelor



Proiect

- ▶ Proiectul reprezinta *un ansamblu de activitati* cu anumite caracteristici si complexitate, pentru care organizatia economica aproba/autorizeaza realizarea proceselor specifice de management de proiect.
- ▶ **Caracteristicile activitatilor:** relativa unicitate, durata determinata de timp, orientarea spre rezultate bine stabilite, limitare a resurselor alocate, risc ridicat etc.
- ▶ **Complexitatea activitatilor:** exprimata prin metrii legate de durata activitatilor, bugetul necesar implementarii, tehnologia utilizata, organizarea necesara, eterogenitatea resurselor etc.



Proiect (definitie ISO 21500/2012)

- ▶ *Definitie:* Proiectul reprezinta un set unic de procese planificate, coordonate si controlate realizate pentru indeplinirea unor obiective.
- ▶ Indeplinirea obiectivelor presupune realizarea unor livrabile (deliverables), respectand multiple constrangeri impuse.
- ▶ Desi multe proiecte sunt similare, fiecare proiect este unic, prin livrabilele oferite, influenta stakeholderilor, resursele utilizate, constrangerile impuse, modul de personalizare a proceselor.

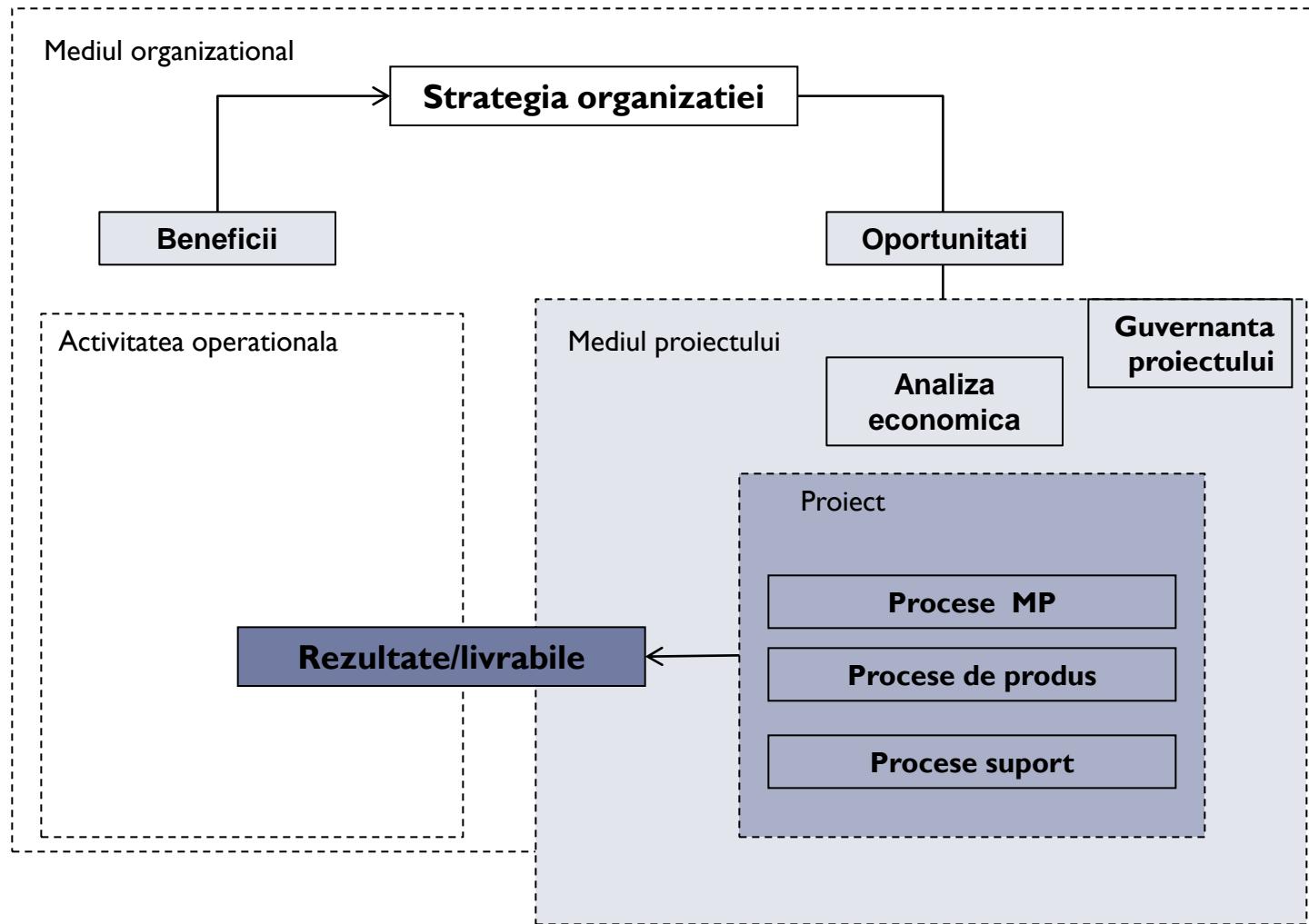


Caracteristicile si complexitatea proiectului

- ▶ Sunt determinate de caracteristicile si complexitatea activitatilor dar nu se reduc la acestea (efectul de sinergie).
- ▶ Proiectul reprezinta o entitate complexa, pentru a carei caracterizare s-au definit numeroase modele.
- ▶ Un model al entitatii proiect rezulta in urma unui proces de modelare a realitatii economice, cu rol explicativ si predictiv.
- ▶ Modelele pot fi normative (definite de **standarde de MP**) sau orientative. O **metodologie de MP** are la baza un model, care este necesar pentru a putea realiza o detaliere a proceselor MP aferente.



Modelul proiectului conform ISO 21500/2012



Initierea proiectelor

Project business model

Challenge	Solution	Unique value proposition	Competitive advantage	Customer segments
The top three customer challenges	The top three features that you provide to solve the customer problem	A single, clear, and compelling message that explains why your solution is unique and worth buying	A unique set of strengths that cannot be easily copied	Target customers
Key metrics		Sales channels		
The top three activities that you measure and their measurable goals		The paths used to reach customers		
Operating costs		Revenue streams		
The fixed and variable costs required to grow the business		The ways that you make money		



Exemplu

Description		Key objectives		
Fredwin Cycling is a freemium mobile fitness cycling application. The app is free to download but users can upgrade to access premium features for \$9.99/mo. Currently, 32% of Fredwin Cycling users have upgraded to a paid subscription.		<ul style="list-style-type: none"> • Increase revenue YoY [+30%] • International expansion [30% of customers outside U.S.] • #1 social fitness cycling app [#1 in the iOS and google play store] 		
Customer targets	Customer challenges	Our solution	Our value	Our pricing
<ul style="list-style-type: none"> • Pro cyclists • Recreational cyclists, MTB and road • Fitness enthusiasts 	<ul style="list-style-type: none"> • Socially connecting with others who ride • Finding trails to ride • Tracking fitness and skill progression 	Social fitness cycling software	We pioneered the market for social fitness and it all started in 2007 with our Fredwin Cycling app. We continue to enhance the mobile app and build out properties that track global cycling races to expand our thought leadership.	Freemium model: <ul style="list-style-type: none"> • Free - Access to the app and all basic features • \$9.99/mo - All basic features plus access to premium features
Our messaging		Go-to-Market		
The #1 social fitness cycling app, built for cyclists by cyclists.		<ul style="list-style-type: none"> • Social channels • App store • Paid search 		
Investment required		Growth opportunity		
<ul style="list-style-type: none"> • \$2MM - New customer acquisition • \$500k - Customer retention • \$1.5MM - Operating costs 		<ul style="list-style-type: none"> • International expansion into EMEA • Premium Plus offerings • Partner expansion 		



Caracteristicile proiectelor

- ▶ Obiective
- ▶ Stakeholderi
- ▶ Context
- ▶ Resurse
- ▶ Constrangeri
- ▶ Organizare
- ▶ Risc
- ▶ etc
- ▶ etc ...



Project Canvas

PURPOSE

Why should the project be started (challenges / problems to be solved)?

Why is the project meaningful and important? ... and for whom?

How will the project change the future?



BUDGET



How much money do you need?

Think of costs for the
... TEAM (internal and external staff)
... RESOURCES (tools, materials, facilities)

TEAM



Which competencies / people
are needed for the project?

Who is a member of the
... core team?
... extended team?

Who is managing the project?

How much time is needed?
(for each person in the team)

CONDITIONS



Known forces, events, and people
that influence the project:

... what / who is supporting the project?
... what / who is restraining the project?

MILESTONES



Define important stages of progress!

What are due dates for
... intermediate results?
... visible accomplishments incl. early wins?
(important for showing progress)
... crucial decisions?

QUALITY

What makes your project a success?

What will make your CUSTOMER happy?

What are the requirements regarding the
... RESULT of the project?
... MILESTONES of reaching the result?

How should the CUSTOMER be involved and informed?



RESULT



Which problem should be solved?

Which value / advantage will be created?

What are you going to create for your
CUSTOMER? (products, services, knowledge)

What are the main components of this result?

CUSTOMER



Who pays for the project? (sponsor)

Who has the final decision to start
and stop the project? (owner)

Who is impacted by the project result?
(recipient)

In case of multiple customers:
Are there (potential) conflicts?

RESOURCES



What work / meeting facilities are required?

Which tools (incl. software) and materials do you need?

Is it necessary (or desired) to involve external partners / experts?

RISKS



Which uncertain future events
might seriously endanger the project,
in case they will happen?

(If you are sure, a future event will happen, or you
can influence it, plan with it as a CONDITION)

TIME

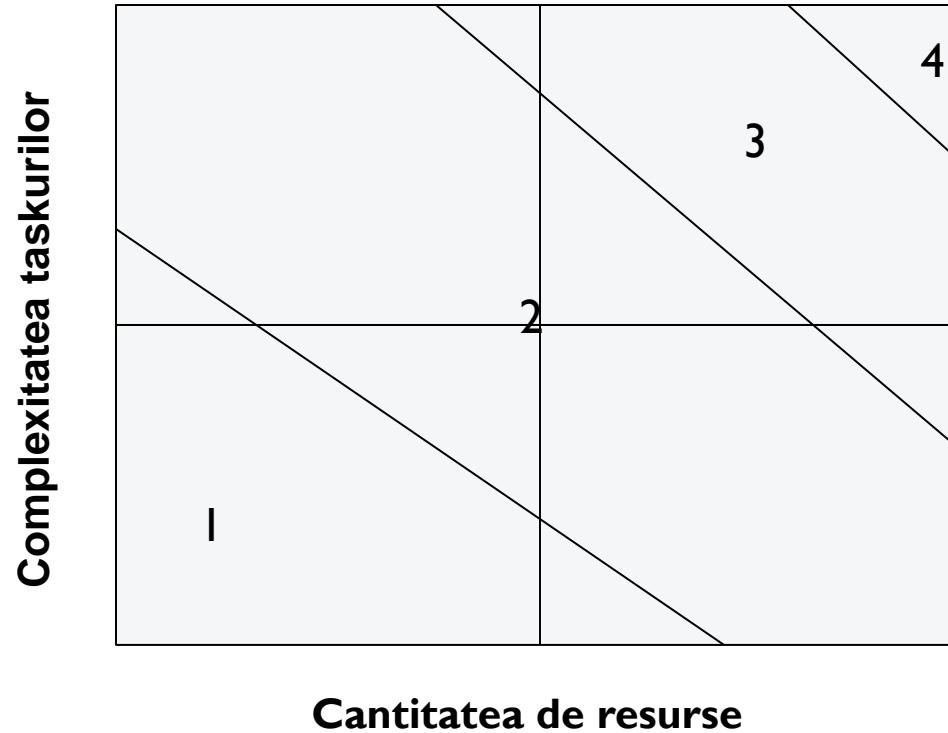


What are the starting dates, i.e. when is the project due to be ... accepted?, ... prepared?, ... kicked off?

What are the end dates, i.e. when is the project due to be ... completed?, ... delivered?, ... liquidated?



Complexitatea proiectelor



Legenda:

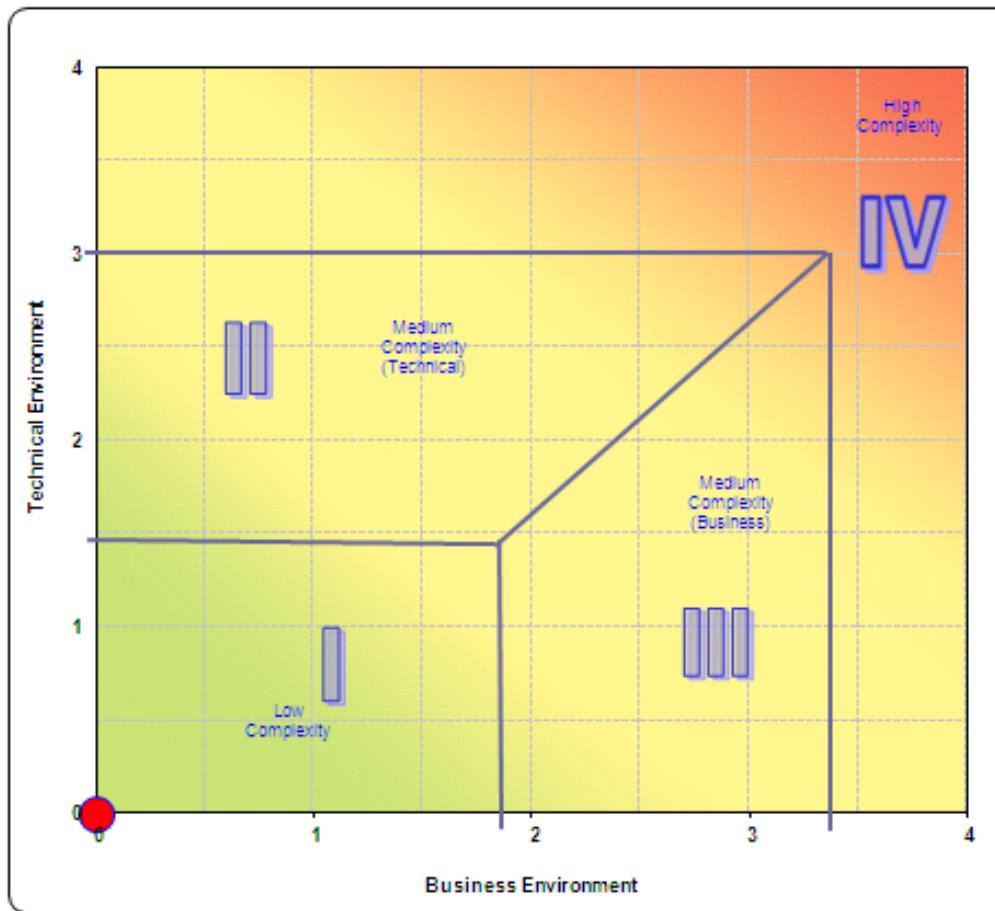
Cantitatea de resurse: un om-luna, un om-an, 10 om-ani

Complexitatea taskurilor: de rutina, combinat rutina-inovative, inovative



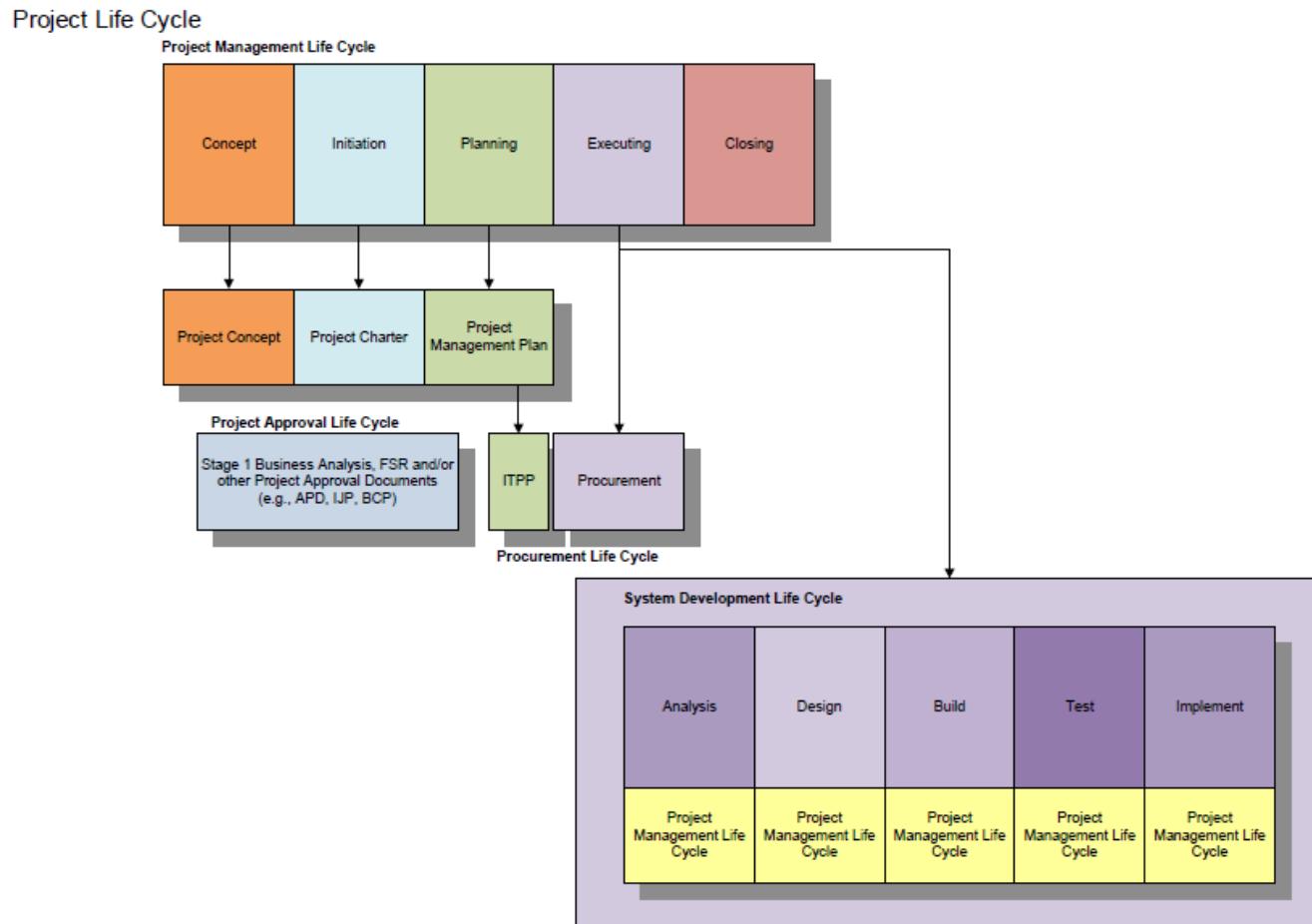
Complexitatea proiectului conform metodologiei CA-PMM

Metodologia CA –PMM este disponibila la adresa:
http://www.cio.ca.gov/Government/IT_Policy/SIMM_17/)



Ciclul de viață al unui proiect

Succesiunea de etape/faze prin care trece proiectul.



Managementul proiectului (MP)

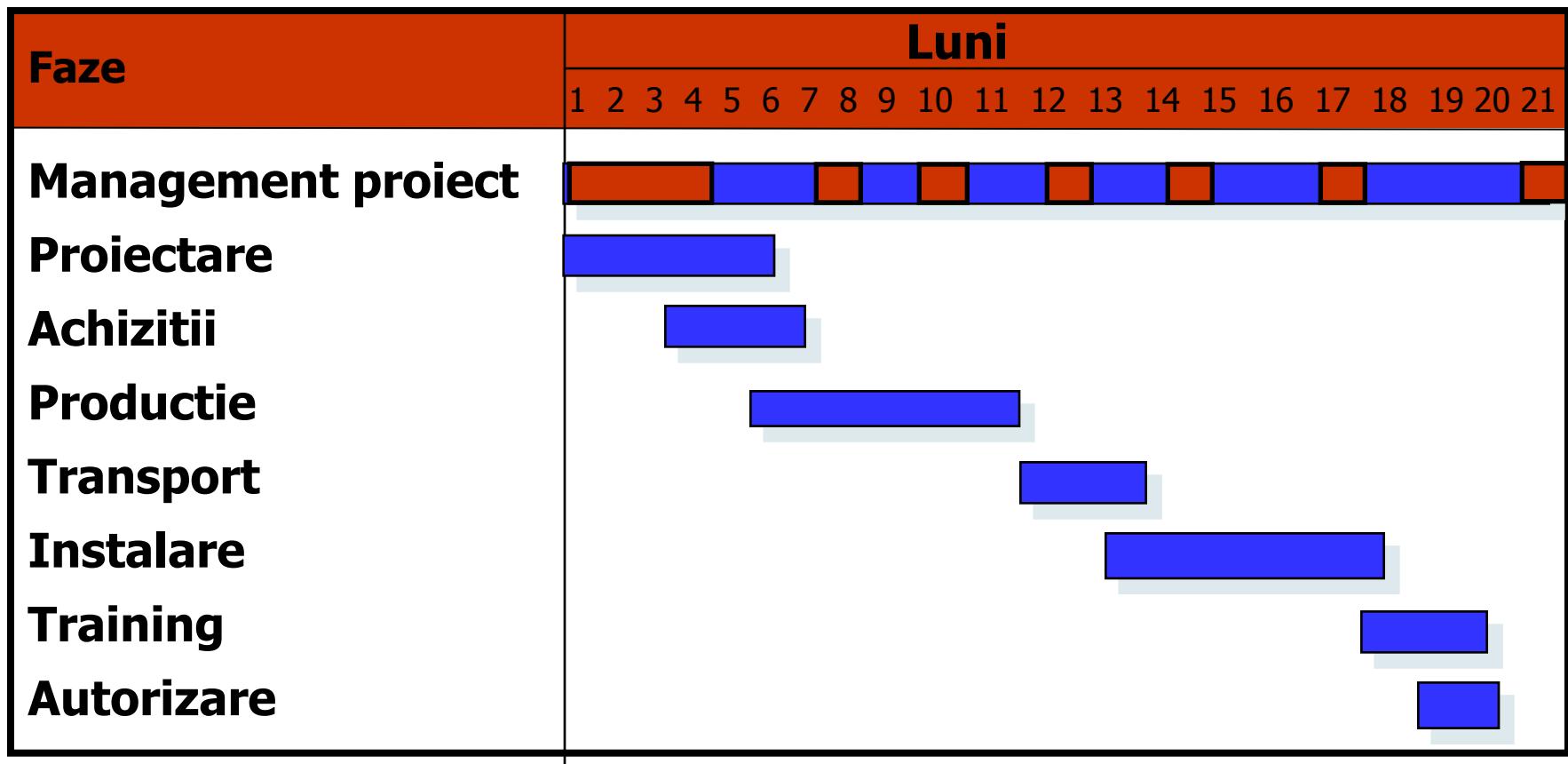
Definitie: Managementul unui proiect (MP) reprezinta aplicarea de metode, instrumente, tehnici si competente pentru executarea/ implementarea unui proiect.

Modelarea MP. Dimensiunile MP

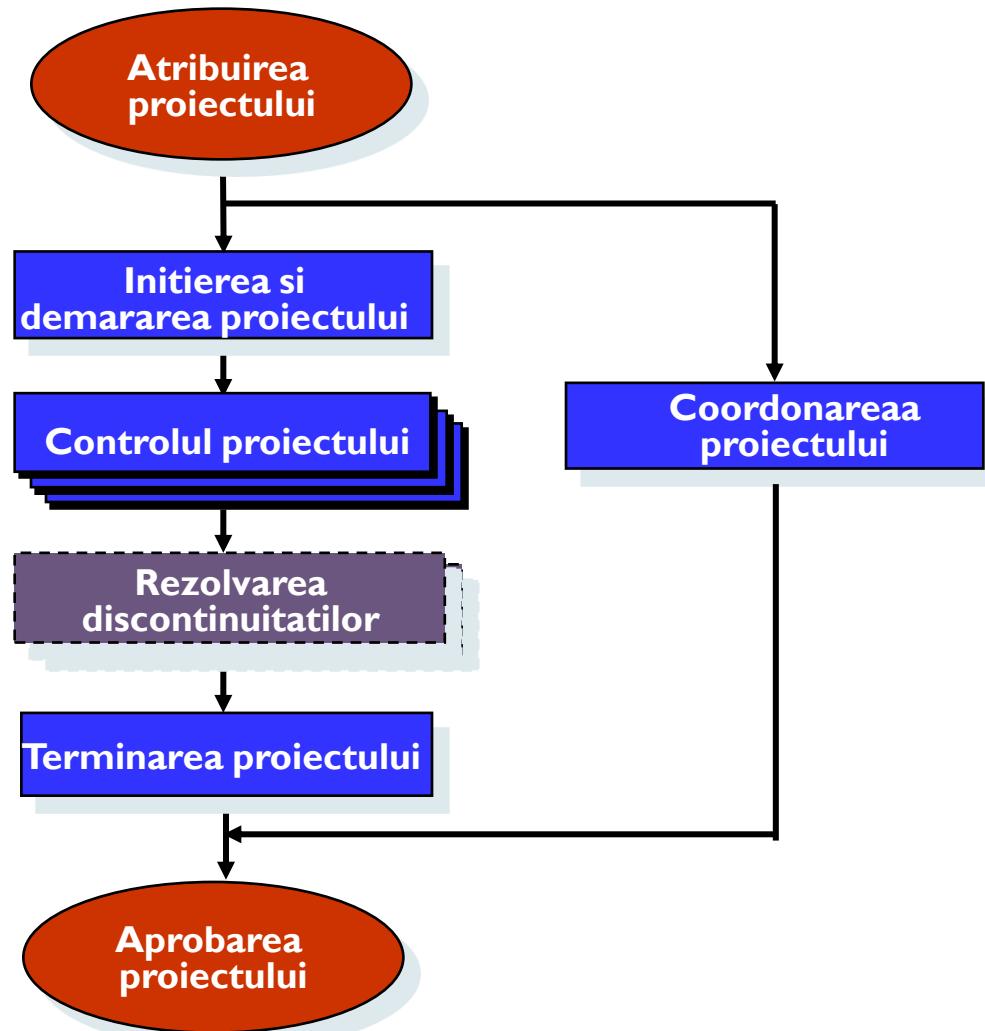
1. Faze si **procese MP**. MP se realizeaza prin procese specifice, reunite in grupuri de procese (faze). Ciclul de viata al managementului de proiect .CAND?
2. **Topici MP** (aspecte vizate, arii de cunostinte) CE?
3. **Metode CUM?**
4. **Roluri MP**. CINE?



MP se realizeaza in paralel cu celealte procese



Grupuri de procese MP



Topici (subiecte, aspecte) ale MP

- Managementul (M) contextului proiectului
- (M) obiectivelor proiectului
- (M) resurselor proiectului
- (M) grupurilor de procese
- (M) organizarii proiectului
- (M) proceselor (detaliat)
- (M) detaliat al resurselor
- (M) executiei (calitate, infrastructura)
- (M) colaborarii in proiect
- (M) informatiilor (sistem informational, intalniri, comunicare, raportare)



Clase de metode de MP

- ▶ Metode de conducere a proiectului (*leading the project*)
- ▶ Metode de analiza a proiectului
- ▶ Metode de planificare generală
- ▶ Metode de planificare detaliată
- ▶ Metode de ancorare a proiectului (gestionare a relațiilor cu mediul)
- ▶ Metode de leadership (comunicare)
- ▶ Metode de monitorizare



Roluri in proiecte

- ▶ Proprietarul proiectului
- ▶ Manager de proiect
- ▶ Echipa proiectului
- ▶ Membrii echipei de proiect
- ▶ Contributorii/participanti la proiect



Program. Managementul programelor

- ▶ Grup de proiecte corelate conduse intr-un mod coordonat (sinergie).
- ▶ Pe langa activitati in proiecte sunt definite si alte activitati (coordonarea lansarilor de versiuni, fabricarea unor componente etc.)
- ▶ Preluarea unor tehnici din managementul functional



Caracteristicile unui program

- ▶ unicitate
- ▶ perioadă determinată mai mare de 1 an
- ▶ definirea de obiective distincte pentru proiectele componente
- ▶ importanță strategică

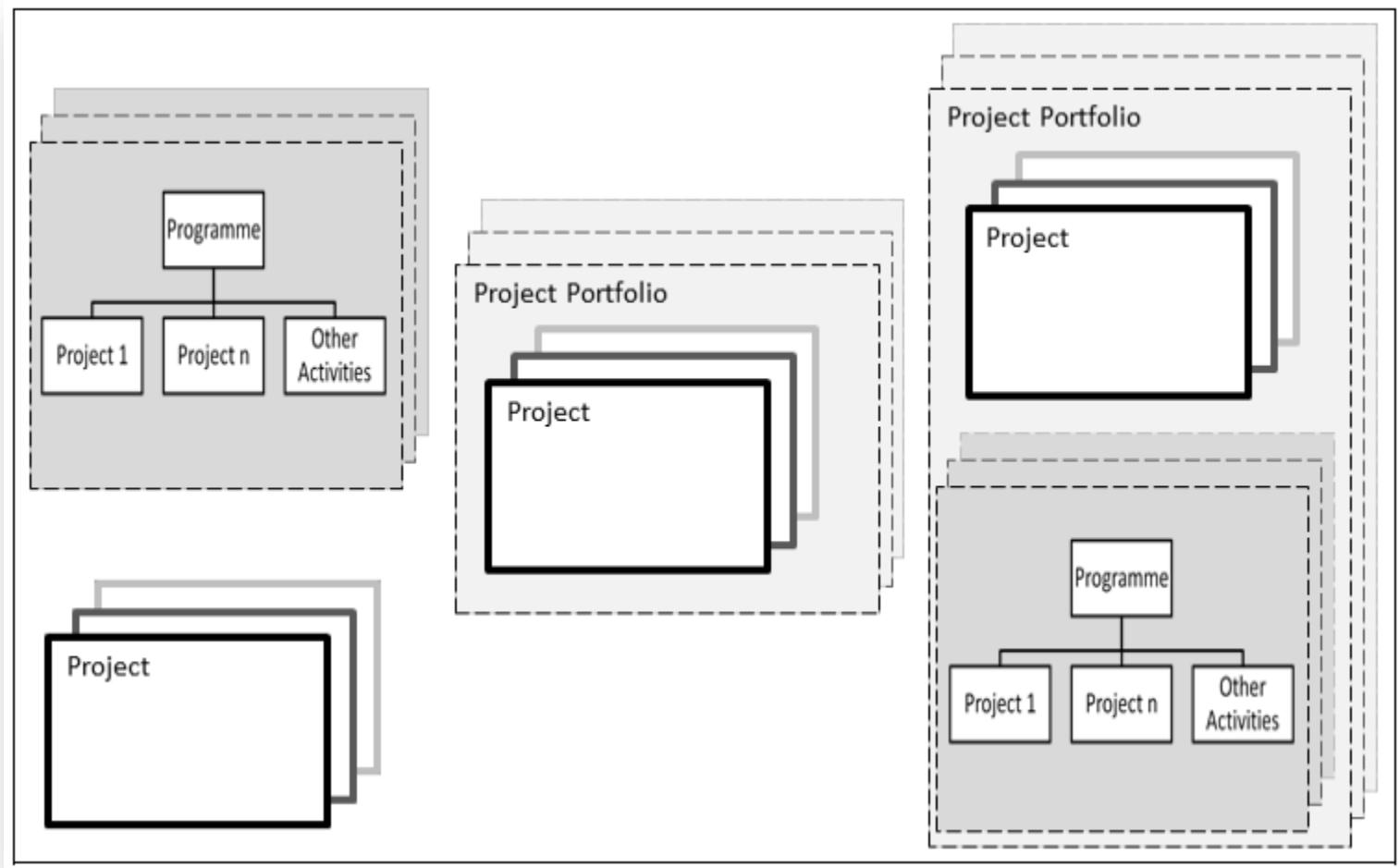


Portofoliu de proiecte

- ▶ Colectia de proiecte sau programe si alte activitati grupate impreuna existente la un moment dat.
- ▶ Pot sa existe sau nu interdependente intre acestea.
- ▶ Managementul portofolului urmareste maximizarea valorii portofolului si presupune: identificarea, prioritizare, autorizare, conducere si control.



Relatia proiecte-programme-portofolii (ISO 21500)



Intrebari

1. Comentati diferențele între următoarele două definitii:
D1: Proiectul reprezinta *un ansamblu de activitati* cu anumite caracteristici si complexitate, pentru care organizatia aproba/autorizeaza realizarea proceselor specifice managementului de proiect.
D2: Proiectul reprezinta *un ansamblu de activitati* cu anumite caracteristici si complexitate, pentru care organizatia **TREBUIE** sa aprobe/autorizeze realizarea proceselor specifice managemenului de proiect.
2. Comentati diferențele dintre caracteristicile activitatilor si caracteristicile proiectului.
3. Prin ce se diferențaza un proiect de un program? Dar un program de un portofoliu de proiecte?



-
- 4. Ce se intelege prin ciclul de viata al unui proiect?
Exemplificati pentru proiectele IT implementate in abordare waterfall si pentru proiectele IT implementate in abordare agila.
 - 5. Ce este managementul de proiect?
 - 6. Care sunt principalele procese de management de proiect?
 - 7. Ce se intelege prin topici (subiecte) ale managementului de proiect?



PARTEA III

Abordari agile in proiectele IT



Ecosistemul abordarilor agile



Istoricul metodologiilor agile

- ▶ Principiile metodologiilor agile isi au originile in Sistemul de Productie Toyota (Toyota Production System - TPS);
- ▶ TPS a fost dezvoltat de Taiichi Ohno si Eiji Toyoda intre 1948 si 1975;
- ▶ TPS a evoluat mai tarziu in Lean Manufacturing;



Manifestul Agile

- ▶ “Noi scoatem la iveală modalități mai bune de dezvoltare de software prin experiență proprie și ajutându-i pe ceilalți. Prin această activitate am ajuns să apreciem”:
 - ▶ **Indivizii și interacțiunea** înaintea *proceselor și uneltelor*
 - ▶ **Software funcțional** înaintea *documentației vaste*
 - ▶ **Colaborarea cu clientul** înaintea *negocierii contractuale*
 - ▶ **Receptivitatea la schimbare** înaintea *urmăririi unui plan*
- ▶ “Cu alte cuvinte, deși există valoare în elementele din dreapta, le apreciem mai mult pe cele din stânga”



Principiile Agile

- ▶ Prioritatea noastră este satisfacția clientului prin livrarea rapidă și continua de software valoros;
- ▶ Schimbarea cerințelor este binevenită chiar și într-o fază avansată a dezvoltării. Procesele agile valorifică schimbarea în avantajul competitiv al clientului;
- ▶ Livrarea de software funcțional se face frecvent, de preferință la intervale de timp cât mai mici, de la câteva săptămâni la câteva luni;
- ▶ Beneficiarii și dezvoltatorii trebuie să colaboreze zilnic pe parcursul proiectului;
- ▶ Construiește proiecte în jurul oamenilor motivați. Oferă-le mediul propice și suportul necesar și ai încredere că obiectivele vor fi atinse;
- ▶ Cea mai eficientă metodă de a transmite informații înspre și în interiorul echipei de dezvoltare este comunicarea față în față;



Principiile Agile – cont.

- ▶ Software funcțional este principala măsură a progresului;
- ▶ Procesele agile promovează dezvoltarea durabilă. Sponsorii, dezvoltatorii și utilizatorii trebuie să poată menține un ritm constant pe termen nedefinit.
- ▶ Atenția continuă pentru excelență tehnică și design bun îmbunătățește agilitatea.
- ▶ Simplitatea - arta de a maximiza cantitatea de muncă nerealizată - este esențială;
- ▶ Cele mai bune arhitecturi, cerințe și design emerg din echipe care se auto-organizează;
- ▶ La intervale regulate, echipa reflectă la cum să devină mai eficientă, apoi își adaptează și ajustează comportamentul în consecință;



Principalele metodologii Agile

- ▶ **Lean software development:**

- ▶ Eliminarea greselilor
- ▶ Amplificarea învătării
- ▶ Decide cât mai târziu
- ▶ Livrarea cât mai repede posibil
- ▶ Împărtăcirea echipei
- ▶ Dezvoltarea integritatii
- ▶ Viziune holistica

- ▶ **Extreme programming:**

- ▶ Programarea în perechi
- ▶ Revizuirea extinsiva a codului
- ▶ Testarea tuturor unitatilor de cod
- ▶ Evitarea programarii functiilor până când acestea sunt vital necesare



Principalele metodologii Agile – cont.

- ▶ **Kanban:**
 - ▶ Se incepe de la procesul existent
 - ▶ Se urmareste realizarea unor schimbari progresive
 - ▶ Se realizeaza alinierea la procese, roluri, responsabilități și titluri, asa cum sunt în prezent
 - ▶ Conducere la toate nivelurile
- ▶ **Scrum-Ban**
 - ▶ O combinatie intre Kanban si SCRUM



Metodología SCRUM



Caracteristici SCRUM

- ▶ Scrum este un cadru pentru dezvoltarea și susținerea produselor complexe.
- ▶ Scrum (subst.): un cadru în care oamenii pot adresa probleme complexe de adaptare, livrând în același timp, în mod productiv și creativ, produse de cea mai mare valoare posibilă.
- ▶ Scrum este:
 - ▶ Usor
 - ▶ Simplu de înțeles
 - ▶ Dificil de stăpânit
- ▶ Scrum este un cadru pentru procese care a fost folosit în managementul dezvoltării de produse complexe încă de la începutul anilor 1990.



Caracteristici SCRUM - cont

- ▶ Scrum **NU** este un proces sau o tehnică de construire a produselor; mai degrabă este un cadru în care poți folosi diferite procese și tehnici. Scrum clarifică eficacitatea relativă a managementului produsului și practicile de dezvoltare astfel încât să se poată face îmbunătățiri.
- ▶ Cadrul SCRUM are la baza:
 - ▶ Roluri (SCRUM Team, SCRUM Master, Product Owner);
 - ▶ Evenimente (sprint planning, sprint review, etc.);
 - ▶ Artefacte (product backlog, sprint, user story, etc.);
 - ▶ Reguli ce leaga totul impreuna;
- ▶ Scrum folosește o abordare iterativă, incrementală cu scopul de a optimiza predictibilitatea și de a controla riscul.
- ▶ Trei piloni susțin orice implementare a controlului procesului:
 - ▶ Transparența;
 - ▶ Inspecția;
 - ▶ Adaptarea;



Valori si principii SCRUM

- ▶ **Focus** – We focus on only a few things at a time, so we deliver valuable items sooner.
- ▶ **Courage** - We feel supported and have all needed resources at our disposal. This gives us the courage to undertake greater challenges.
- ▶ **Openness** - As we work together, we practice expressing how we're doing, and what's in our way. We learn that it is good to express concerns, so that they can be addressed.
- ▶ **Commitment** - Because we have great control over our own destiny, we become more committed to success.
- ▶ **Respect** - As we work together, sharing successes and failures, we come to respect each other, and to help each other become worthy of respect.

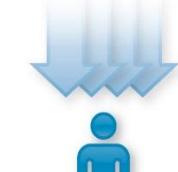


Elemente SCRUM

- Roluri
- Artefacte
- Intalniri

The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Product Owner



The Team



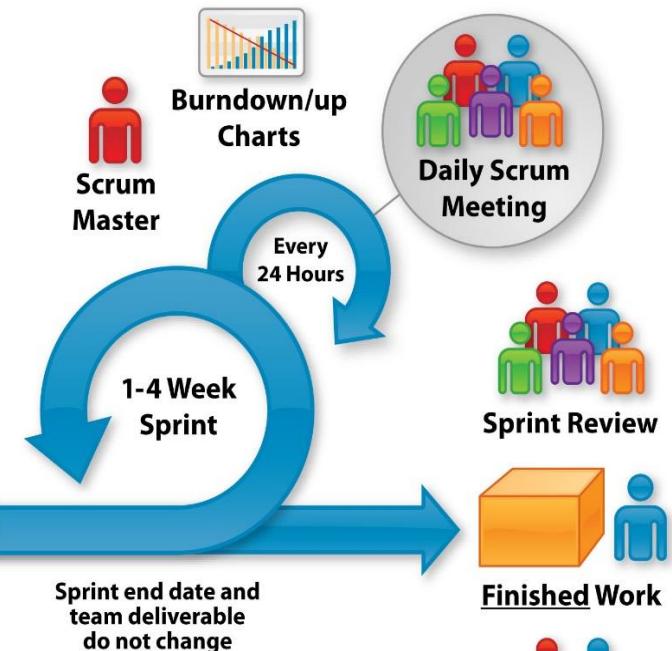
Product Backlog



Sprint Planning Meeting



Sprint Backlog



Roluri SCRUM: Product Owner

- ▶ Product Owner-ul este persoana care intelege cel mai bine viziunea din spatele produsului dezvoltat si valoarea de business a acestuia;
- ▶ Product Owner-ul este interfata intre stakeholders, client, utilizatori si echipa de dezvoltare;
- ▶ Este responsabil sa identifice feature-urile ce trebuie dezvoltate si sa transforme requirement-urile in cerinte clare pentru echipa de dezvoltare;
- ▶ Product Owner-ul este responsabil de maximizarea valorii produsului și a muncii Echipei de Dezvoltare;
- ▶ Product Owner-ul este **singura persoană responsabilă de Product Backlog;**

The Product Owner



The Complete Product Owner | William Gill | <http://williamgill.de>

Roluri SCRUM: Product Owner – cont.

- ▶ Product Owner-ul este o persoană, nu un comitet. Product Owner-ul poate reprezenta interesele unui comitet în Product Backlog, dar cei care vor să schimbe prioritatea elementelor din Backlog trebuie să îl convingă pe Product Owner;
- ▶ Pentru ca Product Owner-ul să reușească, întreaga organizație trebuie să-i respecte deciziile. Deciziile Product Owner-ului sunt vizibile în conținutul și ordinea elementelor din Product Backlog. Nimănui nu îi este permis să îi spună Echipei de Dezvoltare să lucreze pe baza unui alt set de cerințe, iar Echipei de Dezvoltare nu îi este permis să acționeze la ceea ce spune oricine altcineva;
- ▶ Product Owner-ul poate impune **CE** se va face dar nu **CAND** se va face;



Roluri SCRUM : Product Owner – cont.

- ▶ Managementul Product Backlog-ului include:
 - ▶ Exprimarea clară a elementelor din Product Backlog;
 - ▶ Ordonarea elementelor din Product Backlog pentru a realiza cel mai bine obiectivele și misiunile;
 - ▶ Optimizarea valorii muncii produse de către Echipa de Dezvoltare;
 - ▶ Garantarea faptului că Product Backlog-ul este vizibil, transparent și clar pentru toți, și că arată la ce va lucra Echipa Scrum în continuare;
 - ▶ Garantarea faptului că Echipa de Dezvoltare înțelege până la nivelul necesar toate elementele din Product Backlog.
- ▶ Product Owner-ul poate să facă activitățile de mai sus sau poate delega Echipa de Dezvoltare să facă acest lucru. Totuși, Product Owner-ul rămâne responsabil.



Roluri SCRUM : Echipa de dezvoltare

- ▶ Echipa de Dezvoltare este formată din profesionisti care livrează incremente, potențial lansabile pe piață, ale produsului “Finalizat”, la sfârșitul fiecărui Sprint;
- ▶ O echipă formată din 5-9 personae, care sunt localizați împreună din punct de vedere geografic;
- ▶ Echipele de Dezvoltare sunt inter-funcționale și au toate competențele necesare ca echipă pentru a crea un Increment al produsului. Scrum nu recunoaște nici un titlu pentru membrii Echipei de Dezvoltare altul decât Dezvoltator;
- ▶ Echipa de Dezvoltare este structurată și împuternicită de către organizație să-și organizeze și să-și gestioneze munca.



Roluri SCRUM : Echipa de dezvoltare – cont.

- ▶ Echipa de Dezvoltare se auto-organizează. Nimeni (nici măcar Scrum Master-ul) nu poate impune Echipei de Dezvoltare cum să transforme elementele din Product Backlog în potențiale Incrementuri ale funcționalității ce pot fi lansate pe piață;
- ▶ Diferiți membri ai Echipei de Dezvoltare pot avea competențe specializate și zone de focalizare, dar responsabilitatea aparține Echipei de Dezvoltare ca un întreg.
- ▶ Dispune de un grad de autonomie ridicat. Decide singura cum să indeplinească obiectivele asumate;



Roluri SCRUM: Scrum master

- ▶ Scrum Master-ul este o persoana responsabila ca metodologia Scrum să fie înțeleasa și adoptata. Scrum Master-ul face acest lucru prin asigurarea că Echipa Scrum aderă la teoria, practicile și rolurile Scrum.
- ▶ Scrum Master-ul este un furnizor de servicii pentru Echipa Scrum. Scrum Master-ul contribuie la schimbarea interacțiunilor astfel încât să maximizeze valoarea creată de Echipa Scrum.
- ▶ **Serviciile Scrum Master-ului pentru Product Owner:**
 - ▶ Găsirea tehniciilor de gestionare eficace a Backlog-ului Produsului;
 - ▶ Ajuta Echipa de Scrum să înțeleagă necesitatea de a avea elemente clare și concise în Backlog-ul Produsului;
 - ▶ Înțelegerea planificării produsului într-un mediu de dezvoltare empiric;
 - ▶ Asigurarea că Product Owner-ul știe cum să aranjeze Backlog-ul Produsului astfel încât să maximizeze valoarea;
 - ▶ Înțelegerea și practicarea metodelor agile;
 - ▶ Facilitarea evenimentelor Scrum după cum este cerut sau necesar;



Roluri SCRUM: Scrum master – cont.

- ▶ **Serviciile Scrum Master-ului pentru Echipa de Dezvoltare:**
 - ▶ Antrenarea Echipei de Dezvoltare să folosească auto-organizarea și inter-funcționalitatea;
 - ▶ Ajutarea Echipei de Dezvoltare să creeze produse de valoare mare;
 - ▶ Eliminarea impedimentelor din calea progresului Echipei de Dezvoltare;
 - ▶ Protejeaza Echipa de Dezvoltare de distrageri interne si externe;
 - ▶ Facilitarea evenimentelor Scrum după cum este cerut sau necesar;
 - ▶ Instruirea Echipei de Dezvoltare în mediile organizaționale unde Scrum nu a fost pe deplin adoptat sau înțeles;
- ▶ **Serviciile Scrum Master-ului pentru organizatie:**
 - ▶ Directionarea și pregătirea organizației în procesul de adoptare Scrum;
 - ▶ Planificarea implementărilor Scrum în cadrul organizației;
 - ▶ Ajutarea angajaților și a celor implicați să înțeleagă și să adopte Scrum precum și dezvoltarea empirică a produselor;
 - ▶ Cauzarea schimbărilor ce duc la creșterea productivității Echipei Scrum;
 - ▶ Colaborarea cu alți Scrum Master

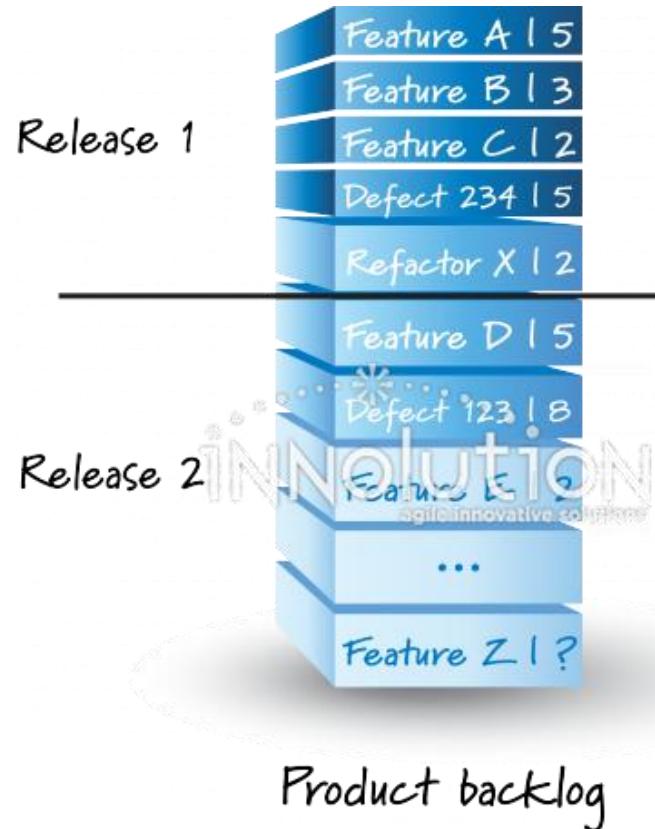
Artefacte SCRUM: Product Backlog

- ▶ Product Backlog-ul este o listă ordonată cuprinzând tot ce poate fi necesar în cadrul produsului, exprimat sub forma unor User Story și este singura sursă de cerințe conținând toate schimbările care trebuie făcute produsului;
- ▶ Product Owner-ul este responsabil de Backlog-ul Produsului, incluzând conținutul său, disponibilitatea sa și ordinea sarcinilor;
- ▶ Product Backlog-ul nu este niciodată complet. Cea mai timpurie dezvoltare a sa prezintă cerințele inițiale, cunoscute și cel mai bine înțelese.
- ▶ Product Backlog-ul evoluează odată cu produsul și mediul în care acesta evoluează. Product Backlog-ul este dinamic, se schimbă constant pentru a identifica ceea ce are nevoie produsul pentru a fi adecvat, competitiv și folositor. Atâtă timp cât un produs există, va exista și Product Backlog-ul acestuia.



Artefacte SCRUM: Product Backlog – cont.

- ▶ Product Backlog-ul listează toate caracteristicile, facilitățile, funcțiile, cerințele, îmbunătățirile și remedierile ce constituie schimbările necesare a fi făcute produsului în versiunile viitoare. Elementele din Product Backlog (User Story) au atribută precum descrierea, ordinea, estimarea și valoarea.
- ▶ Adesea, mai multe Echipe Scrum lucrează împreună la același produs. Un singur Product Backlog este utilizat pentru a descrie munca ce urmează a fi făcută pe produs.



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

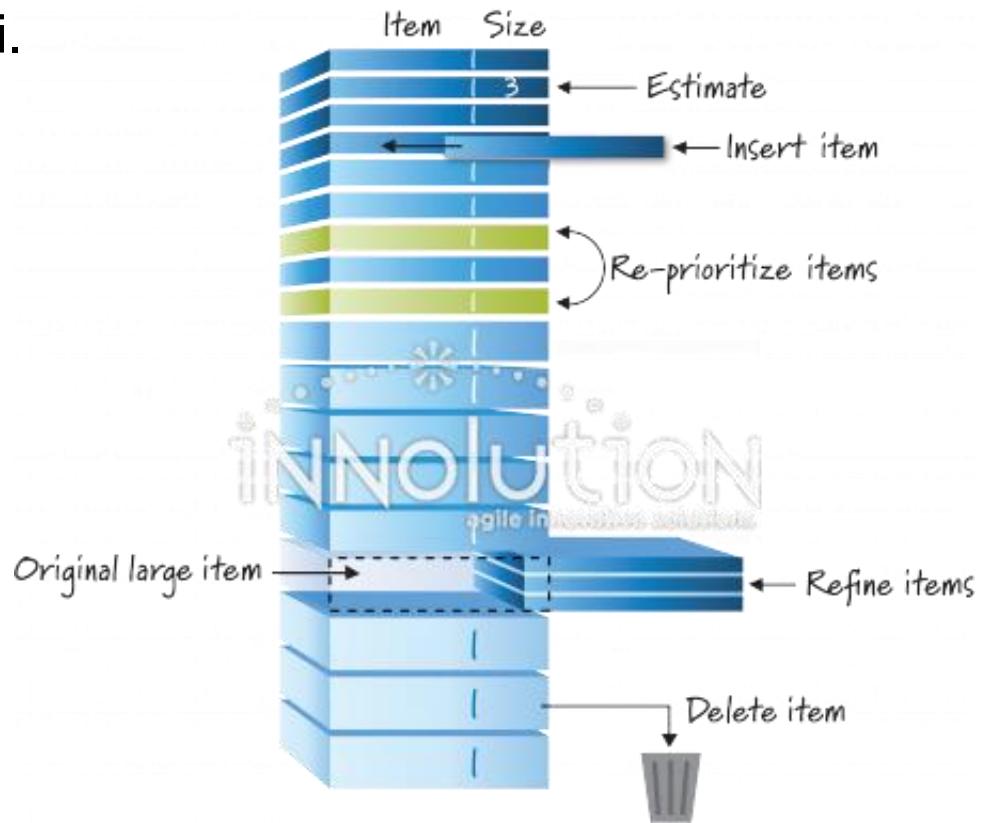


Artefacte SCRUM: Product Backlog – cont.

- ▶ Rafinarea Product Backlog-ului (Backlog Grooming) reprezintă activitatea de adăugare de detalii, estimări și ordonare a elementelor din Product Backlog. Acesta este un proces în continuă desfășurare în care Product Owner-ul și Echipa de Dezvoltare (în practica SCRUM Master-ul) colaborează asupra stabilirii detaliilor din Product Backlog.
- ▶ Elementele din Product Backlog ordonate în susul listei sunt mai clare și mai detaliate decât cele ordonate în josul listei.
- ▶ Elementele din Product Backlog care vor ocupa Echipele de Dezvoltare pentru următoarele câteva Sprint-uri sunt atât de rafinate încât orice element să poată fi “Finalizat” (“Done”) în cadrul duratei unui Sprint.
- ▶ Echipa de Dezvoltare este responsabilă pentru toate estimările. Product Owner-ul poate influența Echipa de Dezvoltare prin a o ajuta să înțeleagă și să selecteze compromisurile, dar membrii echipei care vor lucra efectiv fac estimările finale.



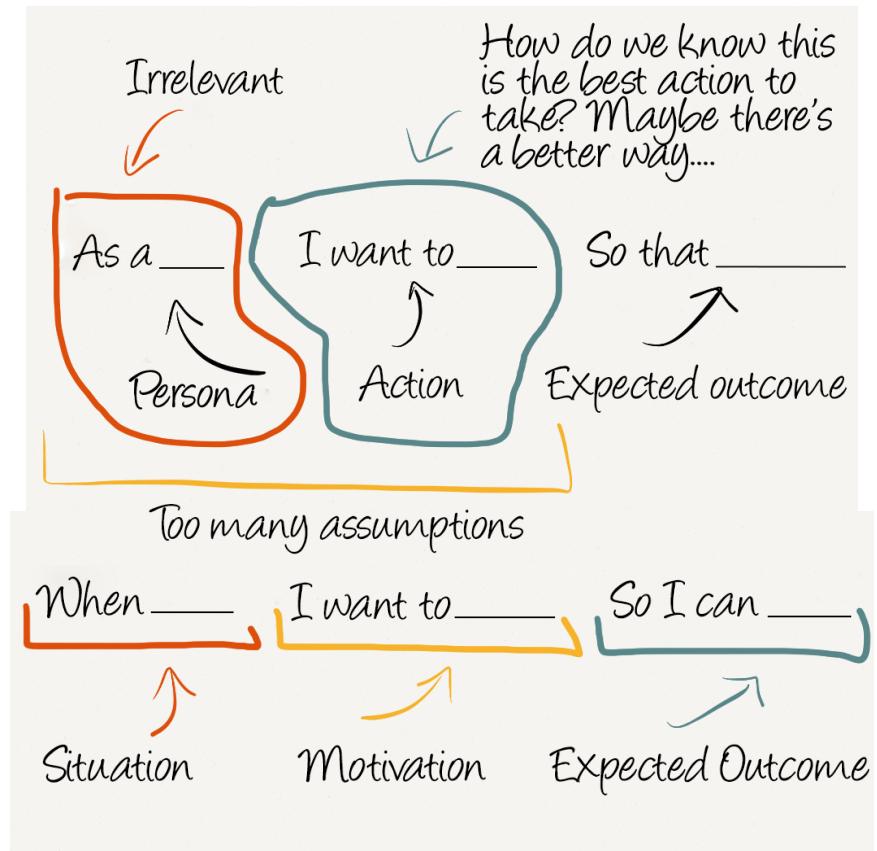
- ▶ Elementele din Product Backlog care pot fi “Finalizate” de către Echipa de Dezvoltare în cadrul unui Sprint sunt considerate “pregătite” (“ready”) sau “gata de acțiune” (“actionable”) pentru selectarea făcută la Planificarea Sprint-ului.



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

Artefacte SCRUM: User Story

- ▶ User Story-ul este o metoda folosita in dezvoltarea software agila pentru a descrie un feature sau o cerinta din punctul de vedere al utilizatorului;
- ▶ As a *<user type>* / want *<requirement>* so / can *<reason>*;
- ▶ Un user story descrie tipul de utilizator, ce doreste acesta si de ce;
- ▶ User Story-urile ajuta la crearea unor descrieri simple pentru un requirement;
- ▶ Formatul este flexibil si poate fi modificat in functie de nevoile sau principiile fiecarei organizatii;



Artefacte SCRUM: Sprintul

- ▶ Inima Scrum-ului este un Sprint, cu o durată fixă (de regula de 3-4 săptamani), în timpul căruia este creat un Increment al produsului, cu statusul "Finalizat" ("Done"), utilizabil și potențial livrabil;
- ▶ Un Sprint nou incepe imediat după încheierea Sprint-ului precedent. Nu există pauze între Sprint-uri;
- ▶ În timpul Sprint-ului:
 - ▶ Nu se aduc modificări care ar putea afecta obiectivul Sprint-ului;
 - ▶ Obiectivele de calitate nu scad;
 - ▶ Scopul poate fi clarificat și re-negociat între Product Owner și Echipa de Dezvoltare pe măsură ce mai multe informații devin disponibile;
- ▶ Fiecare Sprint poate fi considerat un proiect cu un orizont de timp nu mai mare de o lună. La fel ca și proiectele, Sprint-urile sunt folosite pentru a realiza ceva. Fiecare Sprint are o definiție a ceea ce urmează să fie construit, un design și un plan flexibil, care îl vor ghida în procesul de realizare cât și în produsul rezultat.



Artefacte SCRUM: Sprint Backlog

- ▶ Sprint Backlog-ul reprezintă un set de elemente ale Product Backlog-ului selectate pentru Sprint, plus un plan de livrare al Incrementului și de realizare a Obiectivului Sprint-ului;
- ▶ Sprint Backlog-ul reprezintă o prognoză dată de Echipa de Dezvoltare cu privire la funcționalitatea cuprinsă în următorul Increment precum și de munca necesară pentru a livra această funcționalitate într-un Increment "Finalizat";
- ▶ Sprint Backlog-ul este fix la nivelul User Story-urilor;



Artefacte SCRUM: Progress Indicators

- ▶ În orice moment din Sprint se poate însuma munca totală rămasă a activităților din Sprint Backlog. Echipa monitorizează munca rămasă zilnic cu scopul de a estima cât mai realist probabilitatea de realizare a Obiectivului din Sprint dar și pentru a-și gestiona progresul.
- ▶ Gestionarea și monitorizarea progresului într-un Sprint se face prin intermediul sedintelor zilnice de SCRUM, taskboard-uri și burn charts;
- ▶ Taskboard-ul este o metoda vizuala de reprezentare a stadiului în care se află un User Story sau activitate;
- ▶ Burnchart-urile sunt reprezentări grafice a muncii ramase de efectuat în Sprintul current, relative la timpul ramas;

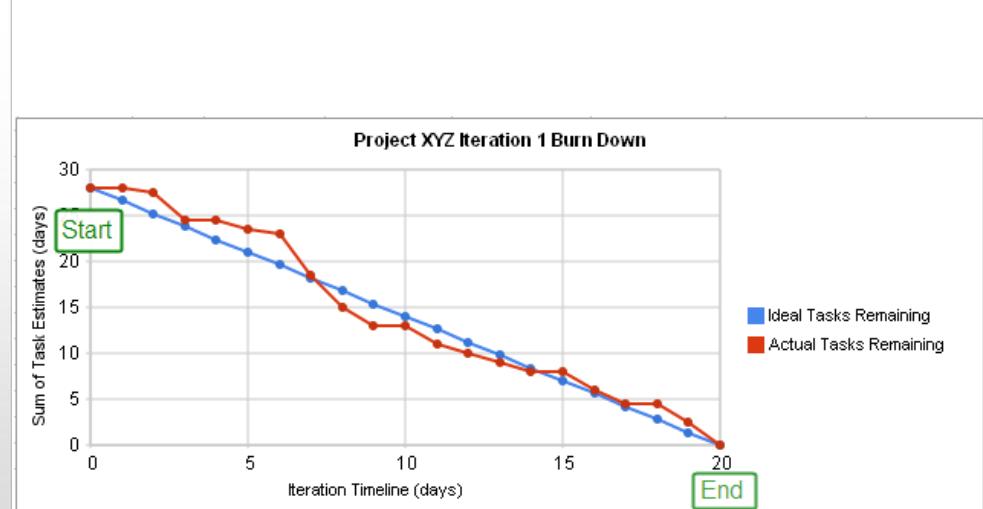


Artefacte SCRUM: Progress Indicators – cont.

Scrum Task Board Template

Company name

Stories	To Do	In Progress	Testing	Done
This is a sample text. Replace it with your own text.	This is a sample text. Replace it with your own text. This is a sample text. Replace it with your own text.	This is a sample text. Replace it with your own text.	This is a sample text. This is a sample text.	This is a sample text. Replace it with your own text.
This is a sample text. Replace it with your own text.	This is a sample text. Replace it with your own text. This is a sample text. Replace it with your own text.	This is a sample text.	This is a sample text. This is a sample text.	This is a sample text. Replace it with your own text.



Artefacte SCRUM: Definition of Done(DoD)

- ▶ Când un element din Product Backlog sau un Increment este descris ca “Finalizat”, toți trebuie să înțeleagă această stare.
- ▶ DoD-ul este folosit de Echipa Scrum pentru a determina când munca este completă pentru Increment.
- ▶ Deși DoD-ul diferă de la o Echipă Scrum la alta, membrii trebuie să aibă o înțelegere comună a ceea ce înseamnă lucrul care urmează să fie terminat, pentru a asigura transparentă.
- ▶ Pe măsură ce Echipa Scrum acumulează experiență este de așteptat ca definiția stării “Finalizat” să se extindă pentru a include criterii mai stricte, cu scopul de a asigura o calitate superioară a Incrementului.
- ▶ Orice proiect sau sistem ar trebui să aibă o definiție a stării “Finalizat” ce este standard pentru orice muncă efectuată.



Artefacte SCRUM: Increment

- ▶ Incrementul reprezintă suma tuturor elementelor din Product Backlog finalizate de-a lungul unui Sprint alături de toate Sprint-urile anterioare.
- ▶ La sfârșitul unui Sprint, noul Increment trebuie să fie în stadiul de “Finalizat”, ceea ce înseamnă că trebuie să fie utilizabil și în concordanță cu ceea ce Echipa Scrum identifică ca fiind definiția stării “Finalizat”.
- ▶ Incrementul trebuie să fie utilizabil și potential livrabil, indiferent de decizia Product Owner-ului de a-l livra sau nu.



Evenimente SCRUM: Sprint Planning

- ▶ Este o intalnire a intregii echipe de SCRUM (Echipa de Dezvoltare, SCRUM Master si PO), limitata in timp (de regula, 1h/saptama de sprint), care are ca scop definirea clara a ceea ce se va face in sprintul urmator si cum;
- ▶ Raspunde la doua intrebari:
 - ▶ Ce va fi livrat in Incrementul care rezulta din Sprint-ul viitor?
 - ▶ Cum va fi efectuată munca necesară pentru ca acest Increment să poată fi realizat?
- ▶ Datele de intrare la această întâlnire reprezintă Backlog-ul Produsului, cel mai recent Increment al produsului, capacitatea proiectată a Echipei de Dezvoltare în timpul Sprint-ului (Team velocity) și performanțele anterioare ale Echipei de Dezvoltare.



Evenimente SCRUM: Sprint Planning – cont.

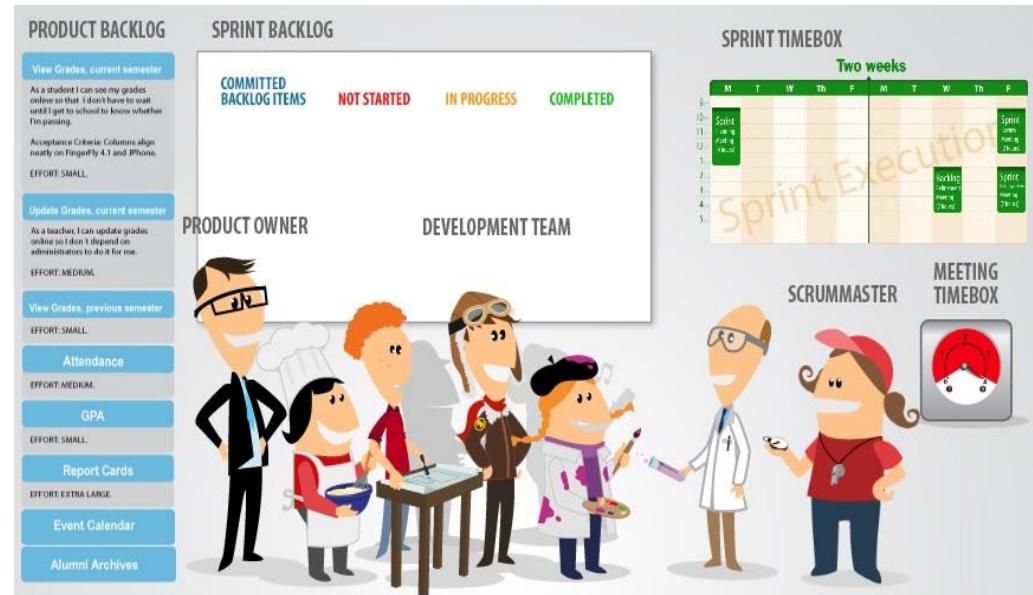
- ▶ Numărul de activități (items) selectate din Backlog-ul Produsului pentru Sprint este numai la latitudinea Echipei de Dezvoltare. Numai Echipa de Dezvoltare poate evalua ce poate realiza în Sprint-ul următor;
- ▶ După ce Echipa de Dezvoltare previzionează elementele din Backlog-ul Produsului pe care le va livra în Sprint, întreaga Echipă Scrum definește obiectivul Sprint-ului;
- ▶ Obiectivul Sprint-ului este un țel care va fi îndeplinit în Sprint prin implementarea Backlog-ului Produsului, și poate oferi îndrumări Echipei de Dezvoltare asupra motivului realizării Incrementului.



Evenimente SCRUM: Sprint Planning – cont.

▶ Structura:

1. Alocarea resurselor;
2. Selectarea US-urilor din backlog;
3. Detalierea US-urilor in subtask-uri;
4. Estimarea US-urilor;
5. Stabilirea obiectivelor sprint-ului;
6. Commitment-ul;



Evenimente SCRUM: Daily SCRUM

- ▶ Este o sedinta ce se desfasoara zilnic pe parcursul Sprint-ului, limitata in timp (nu mai mult de 15 minute), la care participa Echipa de Dezvoltare si SCRUM Masterul, care are ca scop sincronizarea activitatilor si stabilirea unui plan pentru urmatoarele 24 de ore;
- ▶ Se desfasoara in fiecare zi, la aceeasi ora, in aceeasi locatie;
- ▶ Nu este o sedinta de raportare a status-ului;
- ▶ Raspunde la trei intrebari:
 - ▶ Ce am facut ieri?
 - ▶ Ce voi face azi?
 - ▶ Ce ma impiedica sa ating obiectivul Sprint-ului?
- ▶ Aceasta întâlnire cheie este una de inspecție și de adaptare.



Evenimente SCRUM: Sprint Review

- ▶ Este o sedinta ce se desfasoara la sfarsitul Sprint-ului, limitata in timp (de regula, 1h/saptamana de Sprint), la care participa Echipa de SCRUM dar si alti invitati (client, utilizatori, stakeholders, etc.) cu scopul de inspecta rezultatele Sprint-ului si Incrementul;
- ▶ Aceasta este o întâlnire informală, nu una de status, iar prezentarea Incrementului este destinată solicitării de feedback și încurajează colaborarea;
- ▶ Product Owner-ul explică care elemente din Product Backlog au fost "Finalizate" și care nu au fost "Finalizate";
- ▶ Echipa de Dezvoltare discută despre ceea ce a mers bine din punct de vedere tehnic în timpul Sprint-ului, problemele de care s-a lovit și modul în care acele probleme au fost soluționate;
- ▶ De asemenea, demonstrează noile feature-uri și răspunde la întrebări cu privire la Increment si identifica oportunitati sau riscuri cu privire la produs si modifica Product Backlog-ul in concordanta;



Evenimente SCRUM: Sprint Retrospective

- ▶ Retrospectiva Sprint-ului reprezintă o oportunitate pentru Echipa Scrum de a se auto-inspecta și de a crea un plan pentru îmbunătățiri care urmează să fie adoptate în cadrul Sprint-ului următor;
- ▶ Retrospectiva Sprint-ului are loc după Revizuirea Sprint-ului (Sprint Review) și înainte de Planificarea Sprint-ului următor (Sprint Planning) și are o durată fixă (de regula, 45 min/săptamana de Sprint);
- ▶ Scopul Retrospectivei Sprint-ului este:
 - ▶ De a inspecta modul în care ultimul Sprint a funcționat cu privire la oameni, relații, proces, și instrumente;
 - ▶ De a identifica și ordona elementele majore care au mers bine și îmbunătățirile potențiale;
 - ▶ De a crea un plan pentru punerea în aplicare a îmbunătățirilor în modul în care Echipa Scrum își desfășoară activitatea.



Evenimente SCRUM: Sprint Retrospective – cont.

- ▶ Scrum Master-ul încurajează Echipa Scrum să-și îmbunătățească, în cadrul contextului Scrum, procesul său de dezvoltare și practicile sale, pentru ca acestea să devină mai eficiente și mai plăcute în Sprint-ul următor.
- ▶ La sfârșitul Retrospectivei Sprint-ului, Echipa Scrum trebuie să fi identificat îmbunătățirile pe care le va implementa în următorul Sprint. Punerea în aplicare a acestor îmbunătățiri în următorul Sprint reprezintă adaptarea la auto-inspectie a Echipei Scrum însăși.



SCRUM – Integrarea elementelor

The Agile: Scrum Framework at a glance

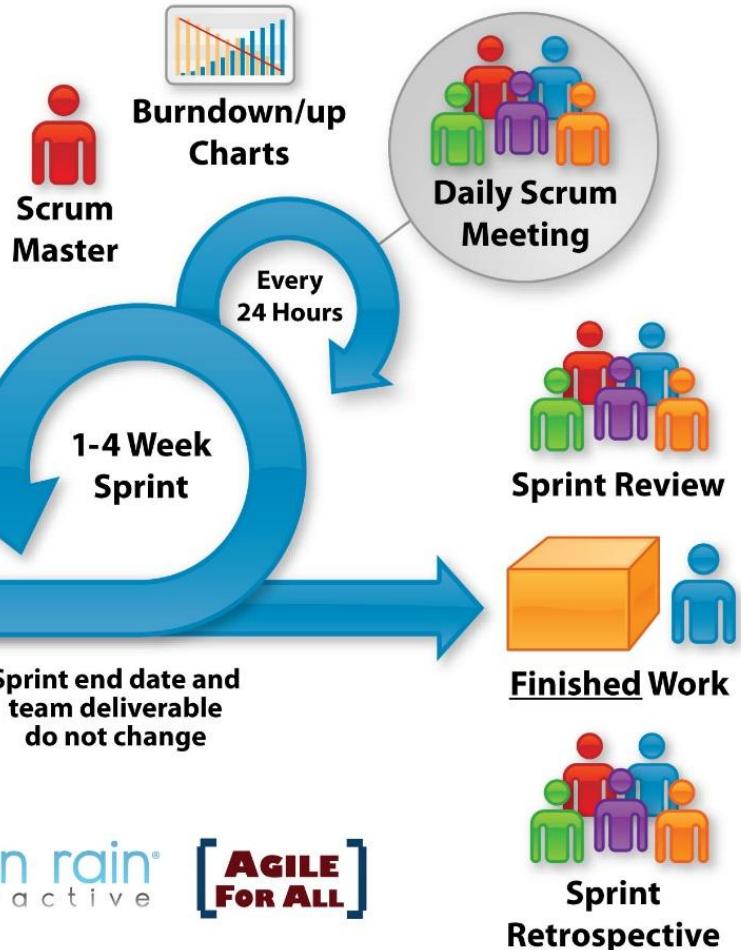
Inputs from Executives,
Team, Stakeholders,
Customers, Users



The Team

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting



Intrebări

1. Care sunt valorile promovate de abordările agile?
2. Care sunt principiile agile? Enumerați minim 3 principii și asociați-le de valorile promovate de abordările agile.
3. Care sunt tipurile de elemente SCRUM?
4. Ce reprezintă un sprint?
5. Ce reprezintă un user story? Cum se redactează/specifică un user story?



Exercitiu

- ▶ Identificati modul in care sunt implementate in *sprintometer* principalele concepte SCRUM, prin:
 - ▶ Realizand o aplicatie simpla in *sprintometer*
 - ▶ Parcurgerea sectiunii VII (incepand cu slide-ul 134)



PARTEA IV

Studiu de caz: Desfasurarea activitatilor intr-un proiect IT implementat in SCRUM



Inaintea primului sprint

- ▶ Alex este asignat **Product Owner** intr-un nou proiect de dezvoltare software.
- ▶ Unul dintre primele sale task-uri este **demararea analizei cerintelor**. Alex incepe sa redacteze cele mai importante use-case-uri si sa le discute cu arhitectii de solutie, reprezentantii clientilor si alti stakeholders. Dupa colectarea use-case-urilor si a cerintelor, de tip *high level*, Alex le include in Product Backlog, sub forma unor **user story-uri**.
- ▶ Alex organizeaza apoi **o sesiune de estimare a duratei/efortului si de prioritizare** a componentelor Product Backlog-ului, cu arhitectii de solutie si cu unii dintre programatorii experimentati. Ca urmare a acestei sesiuni, tuturor componentelor Product Backlog-ului li se asociaza o prima estimare si o prioritate.
- ▶ Alexe incepe apoi **detalierea** componentelor Product Backlog-ului in user stories mai detaliate.
- ▶ Alex convoaca apoi **echipa de dezvoltare** pentru **prima intalnire de planificare sprint** (Sprint Planning meeting).



Seventă/succesiunea activitătilor



Sprint 1 - Ziua 0

Sprint Planning meeting

In timpul intalnirii, **Alex prezinta** componentele Product backlog-ului, incepand cu cele cu prioritate mare. **Echipa** solicita clarificari in cazul in care exista aspecte neclare (*open issues*). Pentru fiecare componenta (item), echipa discuta si redefineste **estimarea marimii**, daca este necesar, stabileste daca are **capacitatea** si expertiza necesare pentru executie, precum si daca sunt disponibile si alte elemente identificate ca fiind necesare.

Dupa aceasta discutie, **echipa se angajeaza** sa execute in cadrul sprintului 1 story-urile: 1, 2, 3, 6, 7 si 8. Echipa considera ca item-urile 4 si 5 nu pot fi realizate in sprint 1, datorita faptului ca infrastructura tehnica nu este inca pregetata/ disponibila.

Imediat dupa finalizarea Sprint Planning meeting, Nicu (**scrum master-ul**) convoaca echipa pentru **definirea task-urilor**, in functie de modul in care item-urile incluse in sprint urmeaza sa fie executate. Taskurile aferente acestor item-uri sunt scrise pe card-uri si incluse in **Sprint Task Board**.

Apoi, fiecare membru al echipei **selecteaza task-urile** la care va lucra.



Sprint 1 - Ziua 1

De dimineata, intreaga echipa se aduna pentru **Daily Scrum Meeting**. Fiecare participant prezinta, pe scurt, ce a realizat deja, actualizeaza estimarea efortului ramas pentru finalizarea taskului pe de card-ul din Sprint Task board, declara ce intentioneaza sa realizeze in ziua curenta si spune daca a identificat obstacole care il impiedica sa lucreze.

Astazi, unul dintre membrii echipei a spus ca are probleme deoarece are nevoie de o noua licenta pentru unul dintre instrumentele software cu care lucreaza. Nicu verifica daca si alti membri ai echipei au aceeasi problema si se angajeaza sa se ocupe de aceasta problema dupa intalnire.

Dupa 15 minute, membrii echipei de dezvoltare se duc la lucru, iar Nicu actualizeaza **diagrama Sprint Burndown**, dupa care il contacteaza pe furnizorul instrumentului software respectiv, comanda licentele si, dupa primirea lor, le pune la dispozitia membrilor de echipa care au nevoie de acestea.



Sprint 1 - Ziua 2

De dimineata, echipa se reuneste, din nou pentru Daily Scrum meeting. Dupa amiaza, unul dintre membrii echipei constata ca nu este sigur referitor la detaliiile unui user story. Il contacteaza pe Alex si discuta cu el aceste *open issues*, dupa care isi continua lucrul.

Sprint 1 - Ziua 28

Este ultima zi din sprint. Nicu invita echipa pentru **Sprint Review Meeting**. Echipa a instalat deja pe un calculator rezultatul curent al dezvoltarii software (codul dezvoltat pana in prezent).

Alex verifica daca implementarea curenta satisface asteptarile si daca functionalitatatile (feature-urile) sunt documentate in mod corespunzator. Concluziile intalnirii sunt: Story-urile 1,2,6 si 7 sunt finalizate, conform asteptarilor. Story 3 nu a putut fi finalizat si nu a fost prezentat deloc. Story 8 are unele componente care trebuie refacute.

Dupa amiaza, echipa se reuneste pentru **Sprint Retrospective Meeting**. Cu aceasta ocazie, se discuta ce a mers bine in cadrul sprintului care tocmai s-a terminat si ce ar putea fi imbunatatit. Un comentariu este acela ca echipa nu detine suficiente informatii despre arhitectura de ansamblu.

Nicu se angajeaza sa invite arhitectul de sistem pentru o prezentare mai detaliata.



Sprint 2 - Day 1

Alex adauga noi componente in Product Backlog, ca urmare a ultimelor intalniri cu clientii. In plus, Alex adauga in Product Backlog si item-urile referitoare la refacerea lui Story 8, reintroducand si itemul aferent Story 3.

Alex invita echipa la Sprint Planning Meeting, pentru sprint-ul 2. Echipa discuta si se angajeaza sa execute anumite stories, avand si suportul lui Nicu. Dupa aceasta, incepe executia Sprintului 2.



PARTEA V

Tehnici agile in proiecte IT



-
- ▶ Tehnici de definire a user story-urilor
 - ▶ Constituirea echipei de dezvoltare. Cerinte pentru membrii echipei de dezvoltare (ED), Product owner (PO) si scrum master (SM)
 - ▶ Stabilirea velocitatii echipei de dezvoltare
 - ▶ Organizarea Sprint Planning Meeting
 - ▶ Estimarea numarului de story points prin tehnica Planning Poker ® (Scrum Poker)
 - ▶ Tipuri de diagrame burndown
 - ▶ Metoda *Lean Coffee* de organizare a intalnirilor
 - ▶ Prioritizarea cerintelor utilizand tehnica MoSCoW



Definirea user story-urilor

- ▶ Item-urile din Product Backlog sunt scrise adesea sub forma unui user story. Un **user story** descrie pe scurt utilizarea produsului de catre un anumit tip de utilizator. Aceasta descriere contine un nume, o scurta descriere si criteriile de acceptanta pentru a considera acel story ca fiind complet.
- ▶ Avantajul user story-urilor consta in faptul ca ele se concentreaza exact pe ceea ce doreste/ are nevoie utilizatorul, fara sa intre in detalii referitoare la cum se va obtine acest lucru
- ▶ **Formate de specificare** a user story-urilor (doua exemple):

As an [actor], I [want|must] [action] so that [achievement] sau

As an [actor], I [want|must] [achievement]

Actor: Reprezinta 'owner'ul acelui user story. El este un anumit tip de user, de exemplu: Administrator, Client, Utilizator neautentificat, etc.

Action: Exprima ceea ce doreste actorul sa realizeze. Poate fi o actiune obligatorie ("must") sau optionala ("want")

Achievement: Exprima ceea ce doreste actorul sa obtina prin realizarea acelei actiuni (este un rezultat definit din perspectiva actorului).



Exemple

As <customer> I want to <see the catalog of salable items> so that <I can order one of them>

As an <administrator> I want <be able to disable accounts>

As a <trainee> I must <answer all the questions>



Constituirea echipei de dezvoltare

Selectarea membrilor echipei se realizeaza astfel incat sa se asigure urmatoarele **cerinte ale echipei**:

- ▶ Sa fie cross-functională (sa acopere setul complet de abilitati necesare realizarii proiectului)
- ▶ Sa asigure dezvoltarea de programe complet functionale, care sa aduca valoare clientului (nivel ridicat al pregatirii de specialitate)
- ▶ Sa fie stabila
- ▶ Sa fie autoorganizabila
- ▶ Sa aiba o dimensiune redusa, adaptata necesitatilor
- ▶ Membrii echipei sa lucreze, pe cat posibil, in aceeasi locatie
- ▶ Sa isi asume responsabilitatea in comun (responsabilitatea echipei)
- ▶ Sa fie focalizata pe rezultate (pe livrarea de valoare) si pe calitate
- ▶ Sa functioneze onest, intr-o atmosfera deschisa si relaxata



Cerinte pentru membrii ED, PO si SM

Membrii echipei de dezvoltare (ED) trebuie:

- ▶ Sa fie buni profesionisti
- ▶ Sa lucreze rapid si eficient
- ▶ Sa fie motivati, rabdatori
- ▶ Sa accepte schimbarile si sa se adapteze rapid la acestea
- ▶ Sa solicite feedback si sa il utilizeze in mod corespunzator
- ▶ Sa fie orientati pe rezultate
- ▶ Sa isi asume responsabilitati individuale in cadrul unei responsabilitati a echipei
- ▶ Sa acorde incredere celorlalți participanți la proiect (membrilor ED, PO și SM)



Cerinte pentru membrii ED, PO si SM

Product owner (PO)

Este responsabil pentru maximizarea valorii de business livrate clientului. El este cel care accepta sau respinge rezultatele obtinute de ED si contribuie la definirea conditiilor de terminare la nivelul fiecarei componente a Product backlog

PO trebuie:

- ▶ Sa detina cunostintele necesare implementarii proiectului
- ▶ Sa fie responsabil si motivat
- ▶ Sa fie prezent, pe cat posibil, in locatia in care lucreaza ED
- ▶ Sa motiveze ED si sa sarbatoreasca realizarile/succesul



Cerinte pentru membrii ED, PO si SM

Scrum master (SM)

Este un facilitator al proceselor de executate in proiect. Ajuta echipa sa se auto-organizeze, o protejeaza, elimina obstacolele, escaladeaza problemele atunci cand este necesar

SM trebuie:

- ▶ Sa aiba capacitatea de a mobiliza echipa
- ▶ Sa practice un leadership de tip *Servant leadership*



Stabilirea velocitatii echipei de dezvoltare

Rata de progres asigurata de catre echipa poarta numele de **velocitate**. Aceasta se exprima, de exemplu, prin numarul de story points executate complet in cadrul unui sprint. Numai acele story point-uri care au fost finalizate vor fi luate in calcul. Includerea unor activitati executate numai parcial (de exemplu codificare fara testare) este interzisa.

Dupa cateva sprint-uri, este foarte probabil ca velocitatea sa fie estimata mai realist/corect. De exemplu, daca velocitatea este de 30 story point si cantitatea de efort ramasa de executat este de 155 story point, putem estima ca avem nevoie de inca 6 sprint-uri pentru a executa toate user story-urile din Product backlog.

In realitate, intrarile din Product Backlog se modifica in mod continuu (noi item-uri sunt adaugate, altele sunt actualizate, chiar sterse).

In cadrul diagramei simple Burndown, nu sunt vizibile schimbarile de continut. Pentru a face vizibile aceste elemente, se pot utiliza alte tipuri de diagrame Burndown.



Organizarea Sprint Planning Meeting

Fiecare sprint incepe cu desfasurarea unui Sprint Planning meeting, format din doua sesiuni de planificare:

- ▶ **WHAT-Meeting.** Echipa se angajeaza sa realizeze anumite user story-uri din Product Backlog. Se defineste astfel un **Sprint Backlog**, care contine toate item-urile/componentele care ar putea fi implementate pana la sfarsitul sprint-ului
- ▶ **HOW-Meeting.** User story-urile sunt descompuse in task-uri mai mici si mai concrete. Dupa aceasta, implementarea poate incepe.



Sesiunea WHAT-Meeting

a) Activitatile de pregatire

Sunt realizate, in principal de catre Product Owner-ul si constau in:

- ▶ Definirea scopului sprintului (**Sprint Goal**). Aceasta reprezinta o scurta descriere a ceea ce incearca sprint-ul sa realizeze
- ▶ Alegerea item-urilor relevante din Product Backlog
- ▶ Actualizarea item-urilor selectate, detalierea acestora in user story-uri de dimensiuni mai mici
- ▶ Estimarea marimii (efortului/duratei) item-urilor si prioritizarea lor

Echipa de dezvoltare isi stabileste capacitatea de efort (velocitatea) pentru sprint-ul planificat. Velocitatea poate diferi de la un sprint la altul, datorita unor factori, precum: concedii, zile libere, solicitarea din intalnirile de proiect, alte activitati din perioada sprintului.



b) Activitatile din cadrul intalnirii

- ▶ Product Owner prezinta scopul sprintului si il discuta cu echipa.
- ▶ Echipa de dezvoltare analizeaza, pe rand, fiecare item din Product Backlog, putand revizui estimarea initiala. Se recomanda ca Product owner-ul sa nu se implice direct in estimarea marimii item-urilor.
- ▶ Echipa se angajeaza sa realizeze item-urile pe care le considera ca fiind posibil de executat integral/complet in cadrul sprint-ului. Decizia de angajare se ia in raport de velocitate si de informatiile disponibile despre item-urile din Product Backlog.

c) **La sfarsitul** WHAT Meeting, lista de item-uri pentru care s-a angajat echipa reprezinta baza pentru realizarea sesiunii HOW Meeting.



Sesiunea HOW-Meeting

Scopul acestei sesiuni este de a completa Sprint Backlog, prin identificarea task-urilor concrete necesare a fi executate pentru finalizarea item-urilor din Product Backlog. Task-urile sunt, de regula, de tip: design, implementation, testing si documentare.

HOW-Meeting se poate desfasura dupa WHAT-Meeting sau concomitent cu aceasta.

Dupa identificarea task-urilor, acestea pot fi estimate de catre echipa, prin numar de om-ore sau de story point necesare pentru finalizarea task-urilor.

Se recomanda ca Product owner sa nu se implice direct in estimarea duratei/ efortului pentru task-uri.



Estimarea numarului de story points prin tehnica Planning Poker® (Scrum Poker)

(https://www.scrum-institute.org/Effort_Estimations_Planning_Poker.php)

In general, la estimarea marimii user story-urilor sau a taskurilor, metodologia SCRUM recomanda utilizarea unor metrii mai abstracte decat cea bazata pe om-ore, care exprima timpul/durata de executie a task-urilor. Este important ca toti membrii echipa sa inteleaga in acelasi mod scala de estimare.

Uzual, ca scale de estimare se utilizeaza:

Scala numerica (de la 1 la 10),

Dimensiunea tricourilor (XS, S, M, L, XL, XXL, XXXL)

Secventa Fibonacci (1, 2, 3, 5, 8, 13, 21, 34, etc.)

Sunt posibile si alte progresii, dar uzual este utilizata secventa Fibonacci intrucat aceasta reflecta incertitudinea in estimarea item-urilor de mari dimensiuni. O estimare cu valoare ridicata semnifica de regula faptul ca acel user story nu este intes in detaliu sau trebuie detaliat (descompus) in item-uri mai mici. User story-urile mai mici pot fi estimate mai bine. Ar fi o pierdere de timp sa se discute daca item-ul are marimea 19, 20 sau 25, intrucat el este prea mare.

Acuratetea estimarilor este asigurata de minimizarea influentarii participantilor in procesul de estimare.



Planning Poker® este o marca înregistrată a Mountain Goat Software, LLC.

Elementele de intrare in Planning Poker®

- Lista item-urilor care trebuie estimate
- Pachete de carduri numerotate. Un pachet cuprinde carduri cu numerele lui Fibonacci, inclusive 0, adica: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc.



Pasii de realizare a estimarilor prin Planning Poker®

Se parcurg, pe rand, item-urile/componentele de product backlog. Pentru fiecare, se executa urmatorii pasi:

- (a) PO prezinta item-ul. Membrii echipei discuta/clarifica respectivul item, putand solicita informatii suplimentare/clarificari de la PO. Pentru ca discutiile sa nu dureze prea mult se poate fixa o limita de timp (de exemplu, un minut). Daca limita de timp este atinsa si ED tot nu a inteles item-ul, se renunta la pasul de estimare si se solicita rescrierea lui.
- (b) Fiecare membru al ED alege in mod secret care reprezinta estimarea sa.
- (c) Expunerea publica a cardului, atunci cand fiecare membru al ED a decis asupra estimarii
- (d) Persoanele cu cele mai mici sau mari estimari pot sa isi justifice estimarile
- (e) Se reia de la (b) pana se atinge consensul



Tipuri de diagrame Burndown

Diagrama Burndown este o reprezentare vizuala a efortului/muncii depuse in fiecare zi in comparatie cu rata proiectata de finalizare pentru release-ul current.

Scopul diagramei este de a oferi informatii care sa permita livrarea solutiei in perioada prevazuta.



Diagrama Burndown simpla

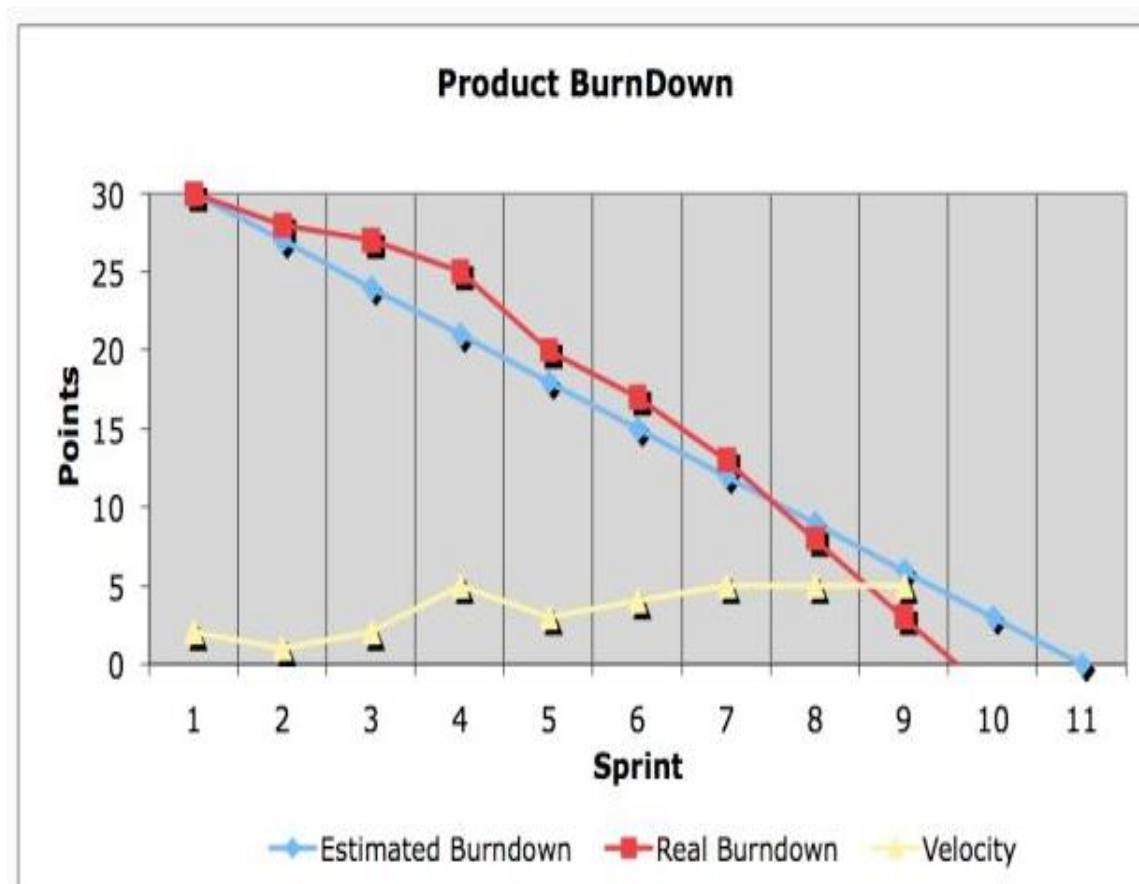
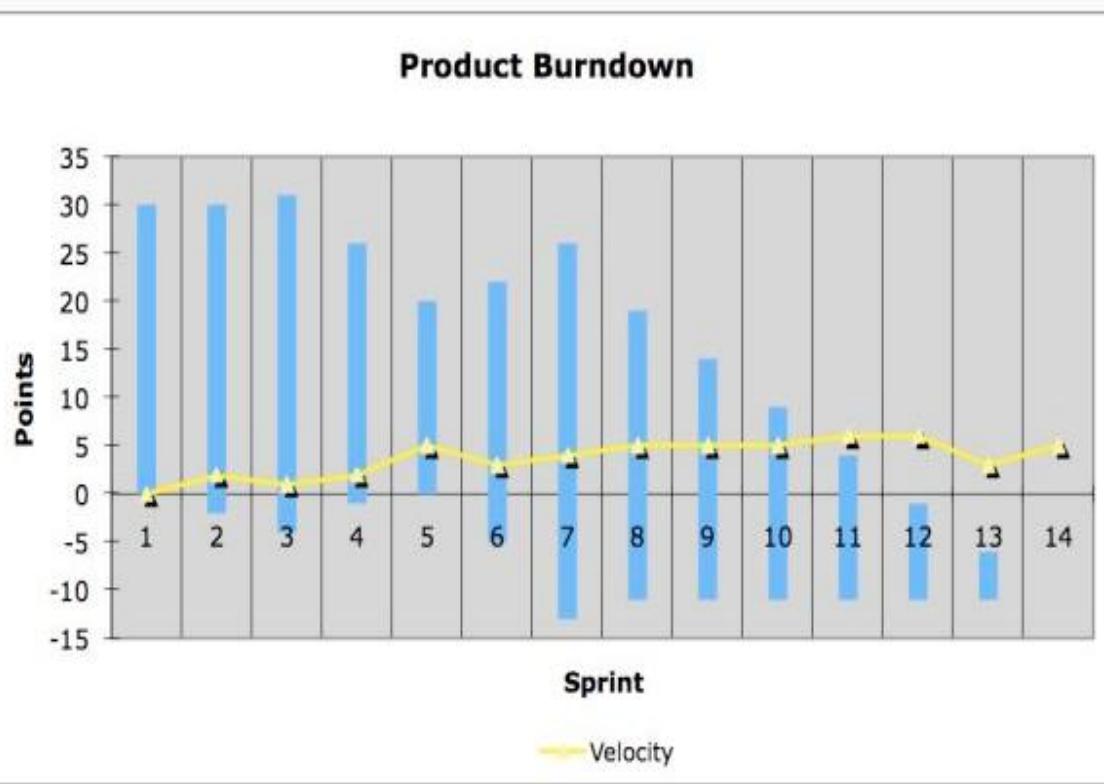


Diagrama burndown cu indicarea schimbarilor in continutul proiectului

Item-urile din product backlog se schimba pe durata proiectului. User story se adauga, altele sunt modificate sau sterse. Este util ca si diagrama burndown sa prezinte aceste schimbari.

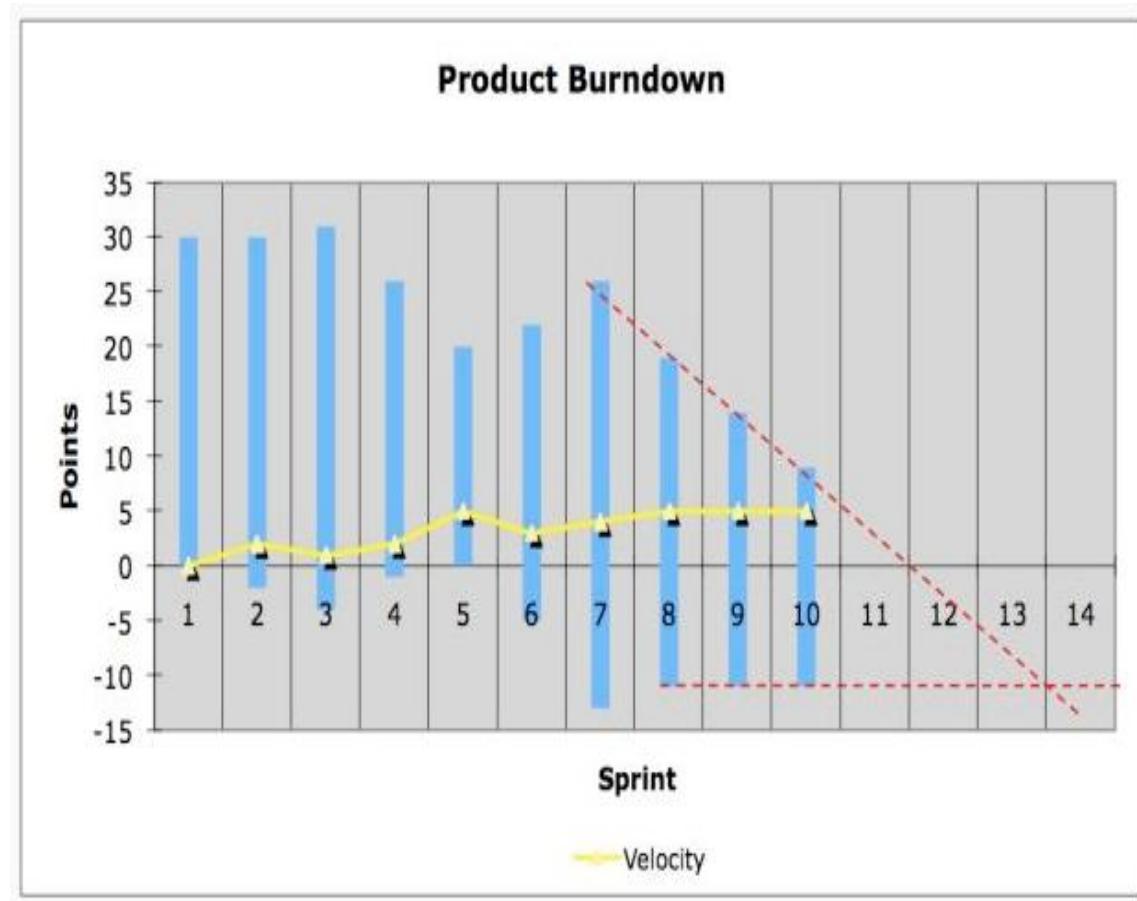


Pentru vizualizarea schimbarilor, munca/efortul ramas a fi depus este prezentat prin bare si nu printr-o linie. Marimea fiecarei bare reprezinta cantitatea de munca ramasa de depus. Velocitatea echipei se scade din partea de sus, iar schimbarile determina modificari in zona de jos a diagramei.



Diagrama burndown cu indicarea ratei de schimbare

Indicarea ratei de schimbare este utila, dar este necesar sa se utilizeze cu precautie aceasta informatie, intrucat tendinta ratei de schimbare nu este aceeasi in timp, la inceput avand tendinta de a fi mai ridicata decat in perioada de finalizare a proiectului.



Metoda Lean Coffee de organizare a întâlnirilor

- ▶ Lean Coffee is a structured, but agenda-less meeting.
 - ▶ Participants gather, build an agenda, and begin talking.
 - ▶ Conversations are directed and productive because the agenda for the meeting was democratically generated.
-
- ▶ Se pretează inclusiv pentru Retrospective meeting, pentru a decide în mod colaborativ agenda și pentru a menține discuția concentrată pe subiect.
 - ▶ Facilitatorul are rolul de a stimula descoperirea intereselor celor implicați în discuție, de a propune subiecte, de a modera discuția și de ghida participanții către ajunge la o concluzie privind subiectele abordate.



Priorizarea cerintelor utilizand tehnica MoSCoW

- ▶ **MoSCoW** (also known as *MoSCoW prioritization* or *MoSCoW analysis*) is a prioritization technique used in software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement.
- ▶ The term *MoSCoW* itself is an acronym derived from the four prioritization requirements categories:
 - ▶ *Must have*
 - ▶ *Should have*
 - ▶ *Could have*
 - ▶ *Won't have but would like.*
- ▶ All requirements are important, but they are prioritized to deliver the greatest and most immediate business benefits early. Developers will initially try to deliver all the *Must have*, *Should have* and *Could have* requirements but the *Should* and *Could* requirements will be the first to be removed if the delivery timescale looks threatened.



-
- ▶ **Must have Requirements** labeled as *Must have* are critical to the current delivery in order for it to be a success. If even one *Must have* requirement is not included, the project delivery should be considered a failure (note: requirements can be downgraded from *Must have*, by agreement with all relevant stakeholders; for example, when new requirements are deemed more important). *MUST* can also be considered an acronym for the **Minimum UsableSubset**.
 - ▶ **Should have Requirements** labeled as *Should have* are important but not necessary for delivery in the current delivery. While 'Should have' requirements can be as important as 'Must have', they are often not as time-critical or there may be another way to satisfy the requirement, so that it can be held back until a future delivery



-
- ▶ **Could have Requirements** labeled as *Could have* are desirable but not necessary, and could improve user experience or customer satisfaction for little development cost. These will typically be included if time and resources permit.
 - ▶ **Won't have Requirements** labeled as *Won't have* have been agreed by stakeholders as the least-critical, lowest-payback items, or not appropriate at that time. As a result, 'Won't have' requirements are not planned into the schedule for the delivery. 'Won't have' requirements are either dropped or reconsidered for inclusion in later deliveries. (Note: occasionally the term **Would like** is substituted for *Won't have*, to give a clearer understanding of this choice).



Exercitii

1. Aplicati tehnicele prezentate in aceasta sectiune pe parcursul elaborarii proiectului de disciplina (vei sectiunea VIII, incepand cu slide-ul 146).

2. Parcurgeti sectiunea VI si precizati care dintre tehnicele prezentate in aceasta sectiune sunt aplicate in proiectele gile ORACLE.



PARTEA VI

Instrumente software pentru implementarea agila a proiectelor IT

JIRA



Crearea unui cont JIRA

<https://www.atlassian.com/software/jira>

- ▶ Se completeaza formularul de Sign in, unde se defineste si numele site-ului.
- ▶ Se declara echipa.
- ▶ Se creaza un proiect.





Jira Software

Cloud Standard

7-day free cloud trial

- ✓ Full access to all features
- ✓ Zero obligation, no credit card needed
- ✓ Add unlimited users
- ✓ Access to sales and technical support

 Looking to host this on your own server?

Create your account

Email *

bodeaconstantanta@gmail.com

[Sign in with a different Atlassian account](#)

First name *

Constanta

Last name *

Bodea

Site name *

simpre2019

.atlassian.net

Yes! I would like to receive promotional emails including product news, events and more from Atlassian and its affiliates.

By clicking below, you agree to the Atlassian [Cloud Terms of Service](#) and [Privacy Policy](#).

Agree and Sign up

NO CREDIT CARD REQUIRED





Great projects need great teams

Add your teammates to plan, track, and release great software together.

Invite your team

constant.a.bodea@ipma.world

Enter their email address

Enter their email address

Let anyone invite other people to join your Jira site

Skip

Next



Recommend a project

Answer a few questions and we will suggest a project type that works best for you and your team.

Help us set up your Jira

I am [new to](#) Jira.

My team is [new to](#) agile methodologies.

We spend our time working on [features](#).

We have a [flexible](#) schedule to finish our work.

Skip

Next



Scrum

Sprint toward your project goals with a modern board and backlog

[Change template](#)[What's in this?](#)

Create project

Name *

Access *

Private

▼ Advanced

Key *



Create



-
- ▶ Se creaza *issue-uri*, pentru diferite sprint-uri.
 - ▶ Pentru fiecare issue, se stabilesc caracteristicile asociate (responsabilitate, marime etc)

simpre-test
Software project

Roadmap

Backlog

Board

Pages

Add item

Project settings

Hi, Constanta Bodea. You're on the new Jira Software experience, which we're adding features to every day.

Take a tour Dismiss

Projects / simpre-test

Backlog

EDU Sprint 1 5 issues

- EDU-1 login page for students
- EDU-2 login page for admin
- EDU-3 define a class schedule
- EDU-4 visualization of classes for a week day
- EDU-5 visualization of class schedule for an entire week

+ Create issue

EDU Sprint 2 4 issues

- EDU-6 define de exams schedule
- EDU-7 define the academic record for a student
- EDU-8 upload the mark for an exam for a student
- EDU Screenshot of the academic record for a student

simpre-test
Software project

Roadmap

Backlog

Board

Pages

Add item

Project settings

Hi, Constanta Bodea. You're on the new Jira Software experience, which we're adding features to every day.

Take a tour Dismiss

Projects / simpre-test

Backlog

EDU Sprint 1 5 issues

- EDU-1 login page for students
- EDU-2 login page for admin
- EDU-3 define a class schedule
- EDU-4 visualization of classes for a week day
- EDU-5 visualization of class schedule for an entire week

+ Create issue

EDU Sprint 2 4 issues

- EDU-6 define de exams schedule
- EDU-7 define the academic record for a student
- EDU-8 upload the mark for an exam for a student
- EDU-9 visulaization of the academic record for a student

Screenshot + Create issue

EDU-1

1

Start sprint

... X

login page for students

STATUS To Do

Description constanta.bodea@ipma.world

ASSIGNEE Constanta Bodea

LABELS None

SPRINT EDU Sprint 1

STORY POINT ESTIMATE 3

The screenshot shows the Jira Software Backlog interface for the 'simpre-test' project. At the top, a purple banner greets the user with their name and encourages them to take a tour or dismiss the message. Below the banner, the 'Backlog' section is displayed under the heading 'EDU Sprint 1'. This sprint contains five issues: 'EDU-1 login page for students', 'EDU-2 login page for admin', 'EDU-3 define a class schedule', 'EDU-4 visualization of classes for a week day', and 'EDU-5 visualization of class schedule for an entire week'. Each issue has a checkbox next to it. Below this, another sprint is listed: 'EDU Sprint 2' with four issues: 'EDU-6 define de exams schedule', 'EDU-7 define the academic record for a student', 'EDU-8 upload the mark for an exam for a student', and 'EDU-9 visulaization of the academic record for a student'. A 'Screenshot' button is located at the bottom of the second sprint's list. On the right side of the screen, a detailed view of the first issue, 'EDU-1 login page for students', is shown. This view includes sections for Status (To Do), Description (constantabodea@ipma.world), Assignee (Constanta Bodea), Labels (None), Sprint (EDU Sprint 1), and Story Point Estimate (3). The 'Screenshot' button from the backlog is also visible here.

-
- ▶ Se starteaza un sprint, declarandu-se durata (in exemplu, durata selectata este de doua saptamani si perioada in care este planificata executia sprint-ului (14-28 octombrie 2019).



simpre-test
Software project

Roadmap

Backlog

Board

Pages

Add item

Project settings

Hi, Constanta Bodea. You're on the new Jira Software experience, which we're adding features to every day.

Take a tour Dismiss

Projects / simpre-test

Backlog

Search CB User

EDU Sprint 1 5 issues

Primele dunctionalitati din sistemul informativ de secretariat CSIE
14/Oct/19 02:53 PM • 28/Oct/19 03:53 PM

13 0 0 Complete sprint ...

- EDU-1 login page for students 3 CB
- EDU-2 login page for admin 5 CB
- EDU-3 define a class schedule 3 CB
- EDU-4 visualization of classes for a week day 1 CB
- EDU-5 visualization of class schedule for an entire week 1 CB

EDU Sprint 2 4 issues

0 0 0 Start sprint ...

- EDU-6 define de exams schedule
- EDU-7 define the academic record for a student
- EDU-8 upload the mark for an exam for a student
- EDU-9 visulaization of the academic record for a student

+ Create issue Screenshot

-
- ▶ La data curentă (13 octombrie 2019), toate issue-urile sunt neințepute. Statusul issue-urilor se poate verifica din facilitatea **Board**.
 - ▶ Alte facilități JIRA: Dashboard, Projects, facilități legate de setări ale contului, de logout.



simpre-test
Software project

Roadmap

Backlog

Board

Pages

Add item

Project settings

Hi, Constanta Bodea. You're on the new Jira Software experience, which we're adding features to every day. [Learn more](#) [Dismiss](#)

Projects / simpre-test

EDU Sprint 1

Primele functionalitati din sistemul informativ de secretariat CSIE

TO DO 5

IN PROGRESS

DONE

+ ADD ISSUE

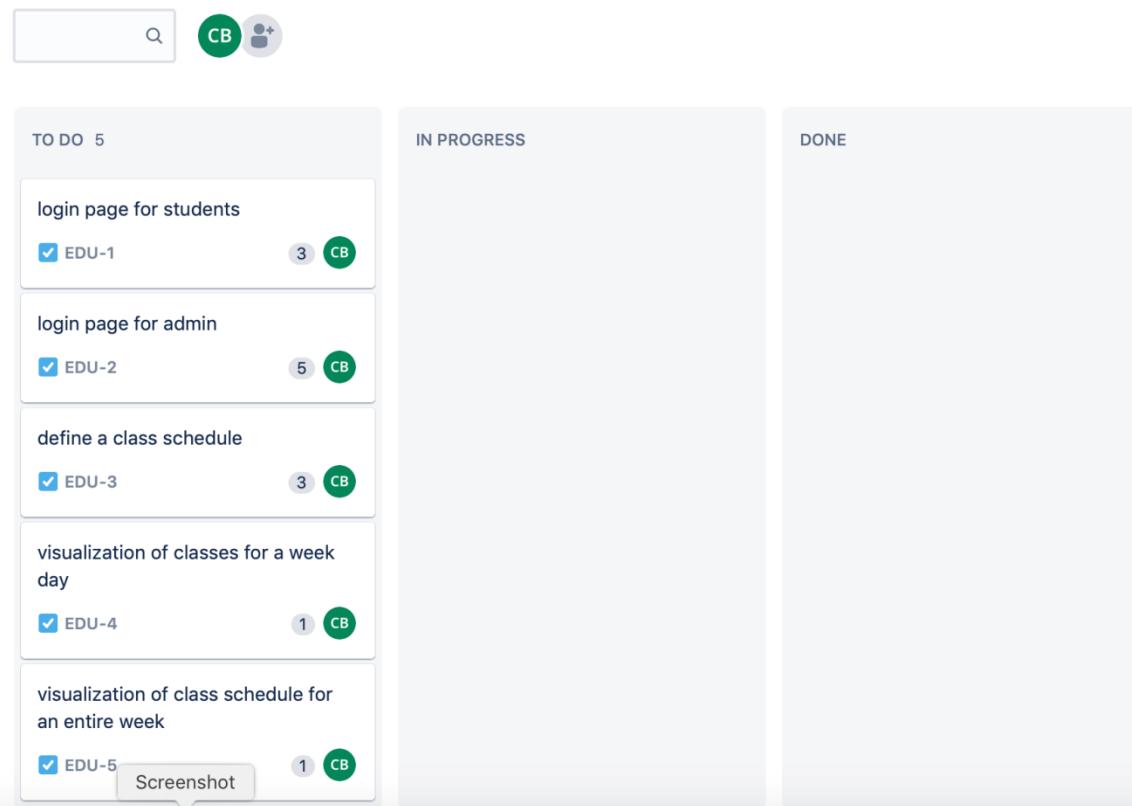
login page for students
 EDU-1 3 CB

login page for admin
 EDU-2 5 CB

define a class schedule
 EDU-3 3 CB

visualization of classes for a week day
 EDU-4 1 CB

visualization of class schedule for an entire week
 EDU-5 1 CB Screenshot



Jira Software

Projects

Create project ▾

Recents

simpre-test Software Project

TRY IT NOW

Set up a customer service desk

EDU board

Requests queue

Q All types

Name ▲

Key ▼

Type

Lead ▼

Constanta Bodea

...

Screenshot

The screenshot shows the Jira Software interface for managing projects. On the left, a sidebar menu includes options like 'Your work', 'Projects' (which is selected), 'Dashboards', 'People', 'Issues and filters', and 'Jira settings'. The main content area is titled 'Projects' and displays a 'Recents' section with two cards: 'simpre-test Software Project' (blue card) and 'Set up a customer service desk' (purple card). Below this is a search bar with a placeholder 'Q' and a dropdown menu set to 'All types'. A table lists one project: 'simpre-test' (Name), 'EDU' (Key), 'Next-gen software' (Type), and 'Constanta Bodea' (Lead). At the bottom, there's a 'Screenshot' button.

Jira Software

Your work

Projects

Dashboards

People

Issues and filters

Jira settings

Default dashboard

Introduction



Welcome to Jira

Not sure where to start? Check out the [Jira 101 guide](#) and [Atlassian training course](#).

You can [customize this text](#) in the Administration section.

Projects

 simpre-test (EDU)

Lead

Constanta Bodea

Screenshot

Assigned to Me

T Key Summary P

EDU-1 login page for students

EDU-2 login page for admin

EDU-3 define a class schedule

EDU-4 visualization of classes for a week day

EDU-5 visualization of class schedule for an entire week

1–5 of 5

Activity Stream

Your Company JIRA



Today



Constanta Bodea updated the Story point estimate of EDU-5 - visualization of class schedule for an entire week

3 minutes ago Comment

Constanta Bodea changed the Assignee to 'Constanta Bodea' on EDU-5 - visualization of class schedule for an entire week

4 minutes ago Comment



Jira Software

- Your work
- Projects
- Dashboards**
- People
- Issues and filters
- Jira settings

Give us feedback

JIRA

Settings

CONSTANTA BODEA

Profile

Account settings

Default dashboard

Introduction

Welcome to Jira



Not sure where to start? Check out the [Jira 101 guide](#) and [Atlassian training course](#).

You can [customize this text](#) in the Administration section.

Projects

simpre-test (EDU)

Lead Constanta Bodea

Screenshot

...

Assigned to Me

T Key Summary P ▾

- | | | |
|-------------------------------------|--|---|
| <input checked="" type="checkbox"/> | EDU-1 login page for students | ↑ |
| <input checked="" type="checkbox"/> | EDU-2 login page for admin | ↑ |
| <input checked="" type="checkbox"/> | EDU-3 define a class schedule | ↑ |
| <input checked="" type="checkbox"/> | EDU-4 visualization of classes for a week day | ↑ |
| <input checked="" type="checkbox"/> | EDU-5 visualization of class schedule for an entire week | ↑ |

1–5 of 5

...

Activity Stream

Your Company JIRA

Today



Constanta Bodea updated the Story point estimate of EDU-5 - visualization of class schedule for an entire week

4 minutes ago Comment

Constanta Bodea changed the Assignee to 'Constanta Bodea' on EDU-5 - visualization of class schedule for an entire week

4 minutes ago Comment

...

Se va continua cu raportarea executiei si cu analiza rapoartelor de progres.

