**Slide 3**

Atunci cand se folosesc date relevante in analiza unui proiect si a felului in care se desfasoara acesta, se pot identifica corect si devreme orice fel de limitari si probleme, care urmeaza a fi rezolvate. De asemenea, se reduc riscurile si se elimina esecurile. Multiple studii au aratat ca, odata cu cresterea timpului de descoperire a unei greseli in cod, creste si costul de rezolvare a acesteia.

Aceste metrici ajuta la identificarea scenariilor atunci cand lucrurile “nu ies la calcul” din cauza unor cerinte de proiect neclare sau conflictuale. Intr-un studiu realizat in 2008, aproximativ 68% din companii erau, din punct de vedere statistic, putin probabil sa aiba succes datorita unor cerinte de proiect slab specificate. Mai exact, exista probabilitatea ca aceste companii sa se incadreze in unul din aceste cazuri:

* sa aiba un buget cu 160% mai mare decat cel original
* sa consume 180% timp in plus fata de cat era asteptat
* sa livreze mai putin de 70% din functionalitatile asteptate ale produsului

**Slide 4**

O metrica software reprezinta o potentiala arie in care masuratori pot fi aplicate eficient asupra unui modul software sau asupra specificatiilor acestuia. In alte cuvinte, o metrica presupune luarea de date din ciclul de viata al dezvoltarii unui proiect si calcularea eficientei dezvoltarii software.

Poate proveni dintr-o singura sau din multiple surse. Ori informatie care poate fi urmarita ar putea deveni o astfel de metrica, desi nu se indicat acest lucru. Ar trebui ca un manager sa ia in considerare toate metricile, datele si indicatorii disponibili? Nu chiar.

De multe ori, managerii cad in capcana de a urmari doar o parte din date, doar pentru ca este usor de obtinut sau este mai usor de interpretat, etc. Motivele pot fi multiple. In realitate, este necesar un echilibru intre cat de usor se poate obtine o metrica, cat de relevant este comportamentul ei in scopul indeplinirii propriilor interese si cat de usoare se poate invata acest comportament.

**Slide 5**

Productivitatea software poate fi definita succint ca fiind raportul dintre functionalitatile software si efortul si costurile consumate in procesul dezvoltarii. Desi exista multiple tipuri de metrici folosite, in majoritatea cazurilor sunt folosite doua tipuri:

* metrici de dimensiune: indica dimensiunea unei caracteristici a produsului. De exemplu, numarul de linii de cod scrise.
* metrici de functionalitate: reprezinta functionaliatatea oferita intr-o perioada fixa de timp.

Metricile de productivitate alese ar trebui sa fie:

* consistente: sa foloseasca definitii clare astfel incat numerele sa aiba si o semnificatie
* disponibile: pot fi folosite pentru comparare
* repetabile: diferite grupuri de utilizatori vor obtine aceleasi rezultate
* credibile: oameni din exterior pot demonstra eficienta masuratorilor

**Slide 7**

Sprint Burndown reprezinta una dintre metricile cheie pentru devoltarea agile scrum.

Un raport burndown comunica complexitatea sarcinilor de-a lungul sprint-ului bazat pe puncte cheie. Scopul echipei este de a termina sarcinile in functie de planificare.

Urmarind aceasta metrica putem obtine detalii importante despre cat de repede sau de tarziu este terminat un sprint comparat cu perioada din planificare. Daca sprint-urile se termina mai repede in mod constant poate reprezenta o lipsa de task-uri, iar daca un anumit sprint se termina in mod constant mai tarziu poate reprezenta o incarcare mai mare de task-uri decat era planificat.

Viteza echipei reprezinta “cantitatea” de software completat intr-un anumit sprint. Poate fi masurata in puncte cheie sau in ore si poate fi folosita pentru estimari si planificari viitoare.

Masurand viteza putem seta asteptari de livrare mai bune si planificari de sprintr-uri mai realiste, putem indentifica daca echipa este blocata, respectiv daca apar anumite challenge-uri care nu au fost prevazute in planificare si putem observa daca diferite actiuni pe care le-am luat in vederea schimbarii procesului de executie au rezultatele asteptate.

Randamentul reprezinta totalul muncii in functie de valoarea acesteia. Este calculat ca fiind sarcinile completate echipa intr-o perioada de timp. Randamentul ar trebui setat in functie de scopurile curente ale business-ului. Daca vrei sa lansezi diferite module fara bug-uri ar trebui ca randamentul sa fie ridicat in ceea ce privesete rezolvarea de bug-uri. De asemenea urmarirea randamentului poate arata atunci cand o echipa este blocata sau cand este supraincarcata.

Durata unui ciclu reprezinta timpul trecut din momentul in care este inceput un task pana cand este finalizat. Este o metrica folosita pentru a estima cat de repede poti livra o anumita functionalitate clientului.

**Slide 8**

Timpul total reprezinta timpul trecut din momentul in care a fost creat un task si pana a fost finalizat. Acesta include durata unui ciclu.

Timpul mediu de reparare reprezinta cat de repede poti livra rezolvarea unui bug consumatorilor.

Acoperirea de cod reprezinta cantitatea de cod testat de catre o echipa de testare.

Rata de bug-uri reprezinta numarul mediu de bug-uri generate de implementarea unei functionalitati noi. Poate ajuta la estimarea valorii codului livrat.

Alte metrici folosite sunt: numarul de erori, utilizarea de CPU/memorie, timpi de raspuns, spatiu folosit pe disc, numar de thread-uri folosite.

**Slide 9**

Numarul de task-uri pe care o echipa le poate completa atunci cand apare o schimbare, comparat cu media estimata pentru o perioada fara schimbare poate reprezenta un mod de a intelege cat de consistent isi termina o echipa sarcinile.

Recidivism-ul - un valoare mare inseamna ca cineva din fluxul de lucru nu a avut acelasi standard ca o persoana de care o urma in flux. Este un indicator bun pentru a identifica cerinte incomplete sau inconsistente.

**Slide 11**

Scopul utilizarii unor metrici si masuratori in ingineria software este de a identifica si controla tot ce poate afecta dezvoltarea software. Mai exact, folosirea unor astfel de metrici contribuie la:

* planificare precisa a proiectului: se poate realiza prin intrebari de genul
  + cat costa fiecare proces?
  + este sau nu este un proces necesar sau eficient?
  + cum putem imbunatati mai departe?
* performanta imbunatatita a dezvoltatorilor software: prin aceste metrici, se cunoaste exact ce performanta este asteptata de la ei. Ei raman implicati intrucat cunosc ce asteptari exista asupra muncii si performantei lor.
* produse mai bune: in urma combinarii acestor factori, produsul rezultat este mult mai bun, datorita capacitatii de a identifica limitarile din timp, de a realiza actiunile necesare pentru a rezolva aceste limitari si de a produce cod mai bun, mai rapid si la un cost mai mic.