Primjena metrika

Šabloni upotrebe

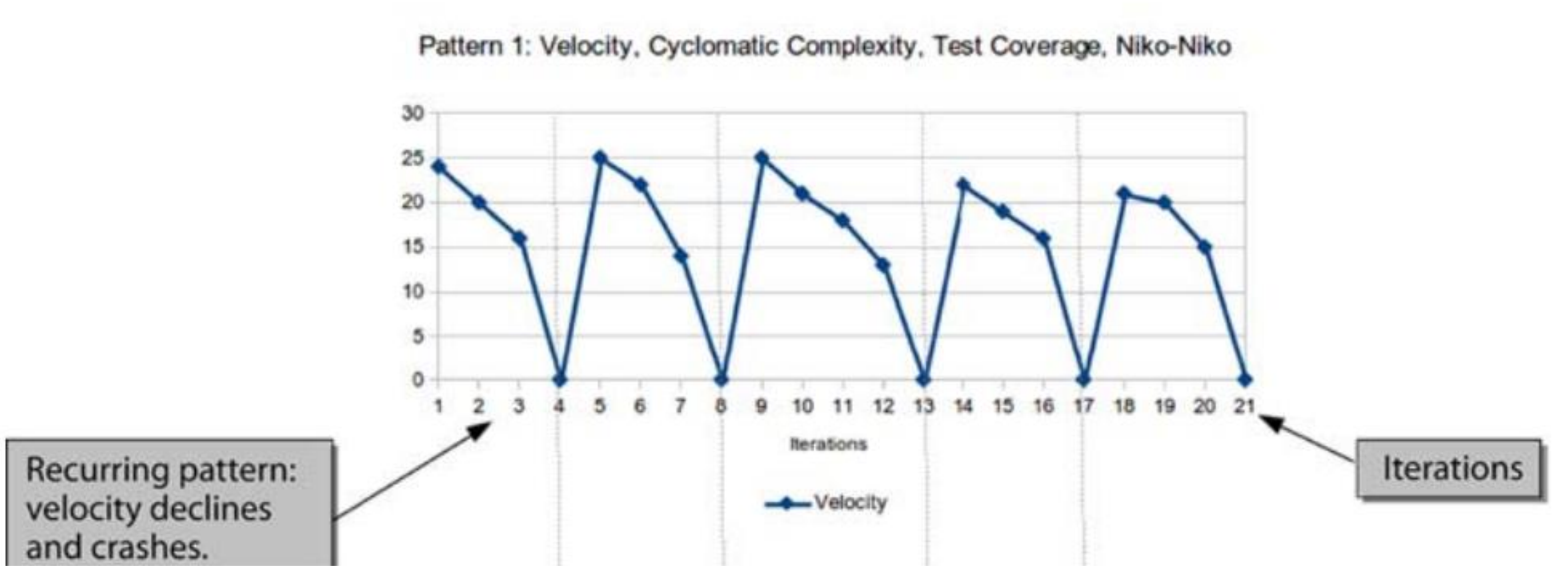
- Više metrika istovremeno da bi izbjegli false positive alarme

- Do sada o metrikama
  - Svrha, funkcija, zavisnosti od metodologije, proces modela i načina isporuke
  - Upravljanje radnim procedurama i prepoznavanje mogućih načina za poboljšanja
  - Tumačenje trendova

Šablon 1

- Primjenjuje se time-box iterativni model, šablon je da tim u prvim iteracijama ima dobre delivery performanse, koje počinju da padaju do situacije u kojoj se cijela iteracija koristi za popravljanje i čišćenje koda umjesto za isporuku production-ready inkrementa
- Koristimo metrike
  - Velocity – količina production ready softvera u iterciji izražena story poenima
  - Cyclomatic complexity – kvalitet koda ocijenjen pomoću broja nezavisnih „grana“ u kodu
  - Automated test average – source code koji prolazi testove
  - Niko Niko calendar – emocionalno stanje tima

Šablon 1 - Velocity

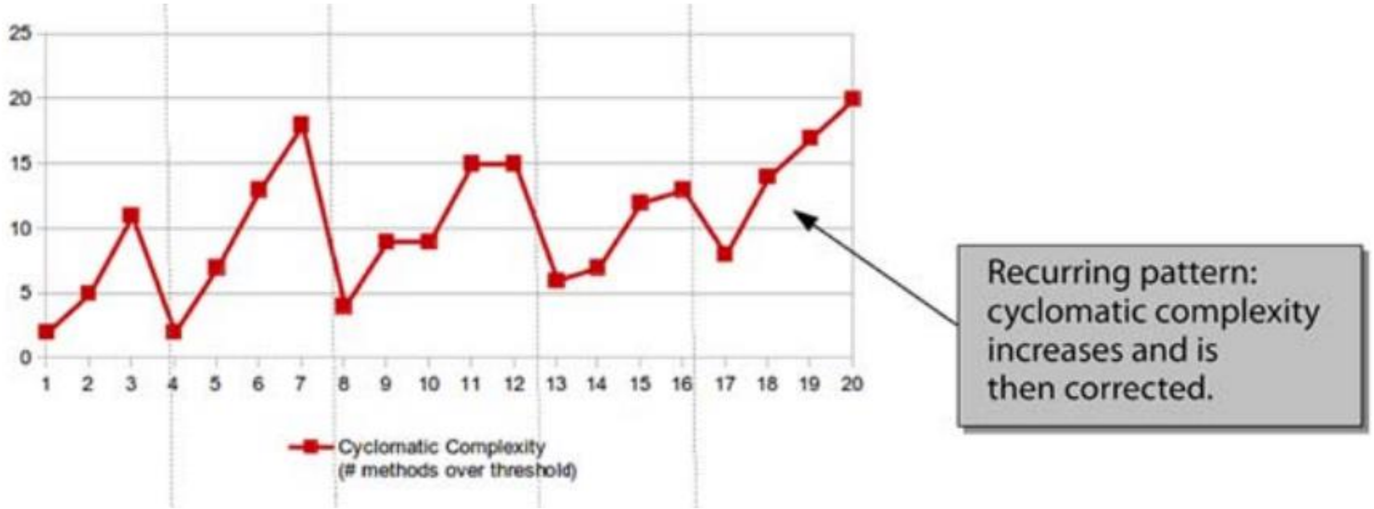


Recurring pattern: velocity declines and crashes.

Iterations

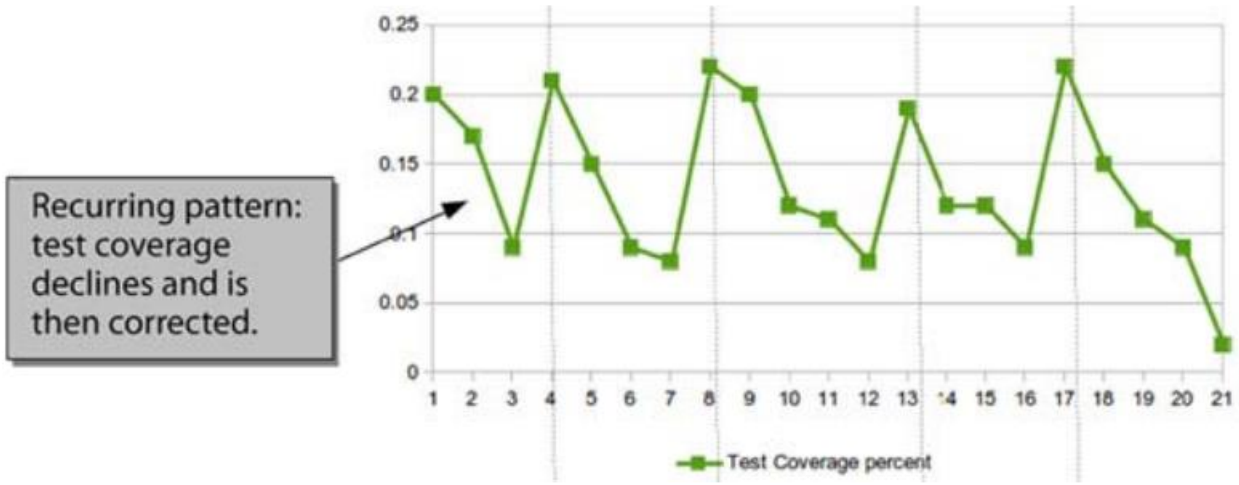
Šablon 1 - Cyclomatic complexity

- *They pile if/else blocks on top of other if/else blocks.*



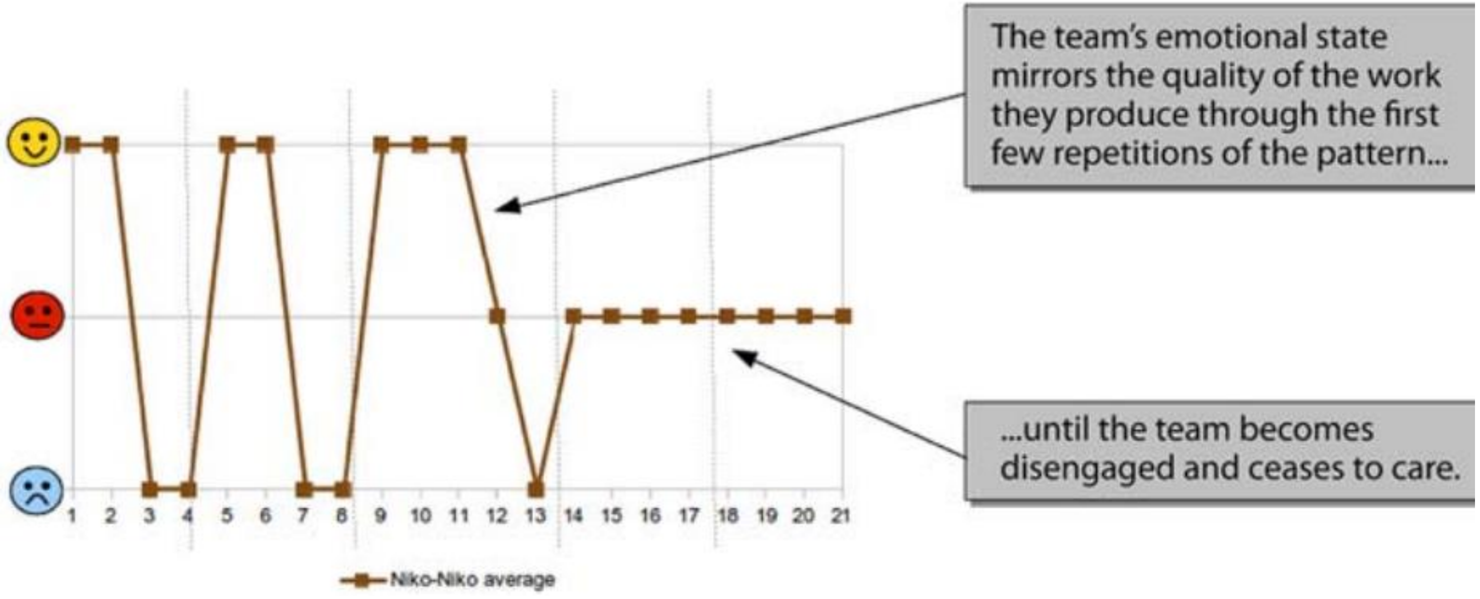
Recurring pattern: cyclomatic complexity increases and is then corrected.

Šablon 1 - Automated test average



Recurring pattern: test coverage declines and is then corrected.

Šablon 1 - Niko Niko calendar



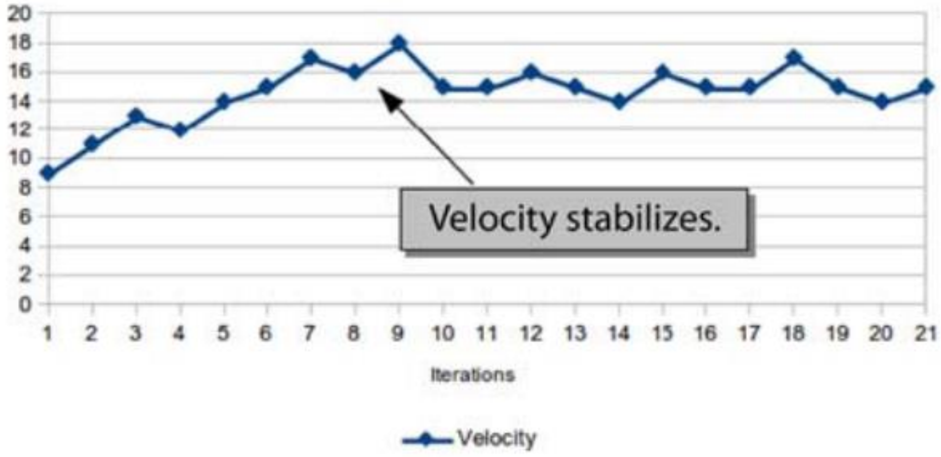
The team's emotional state mirrors the quality of the work they produce through the first few repetitions of the pattern...

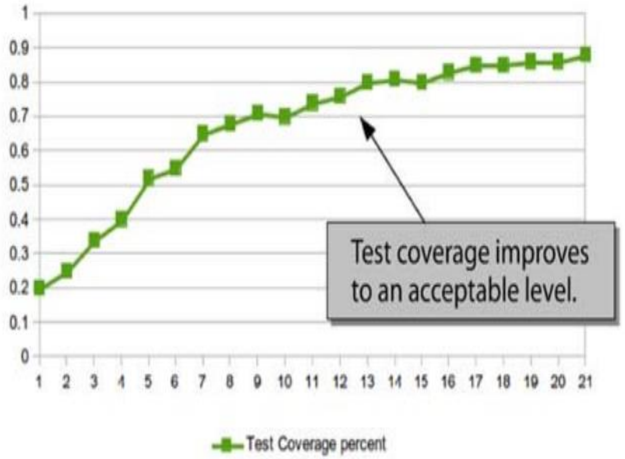
...until the team becomes disengaged and ceases to care.

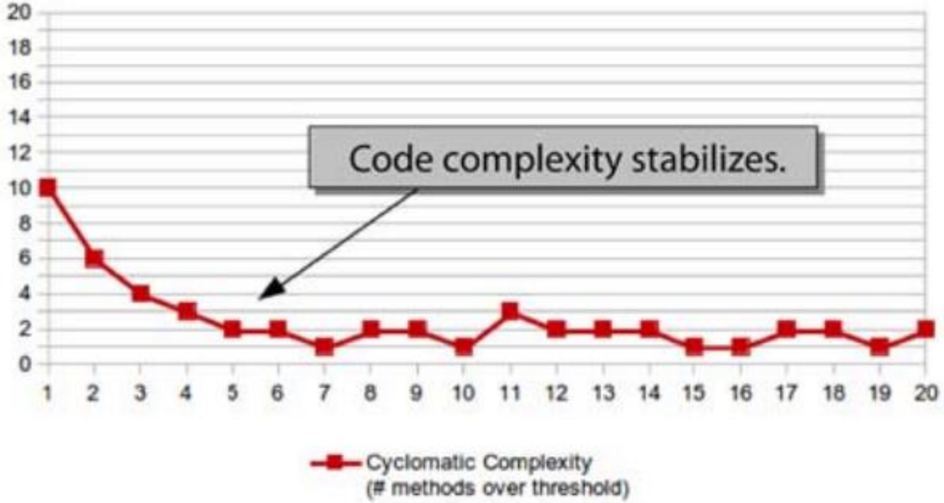
Korektivne akcije

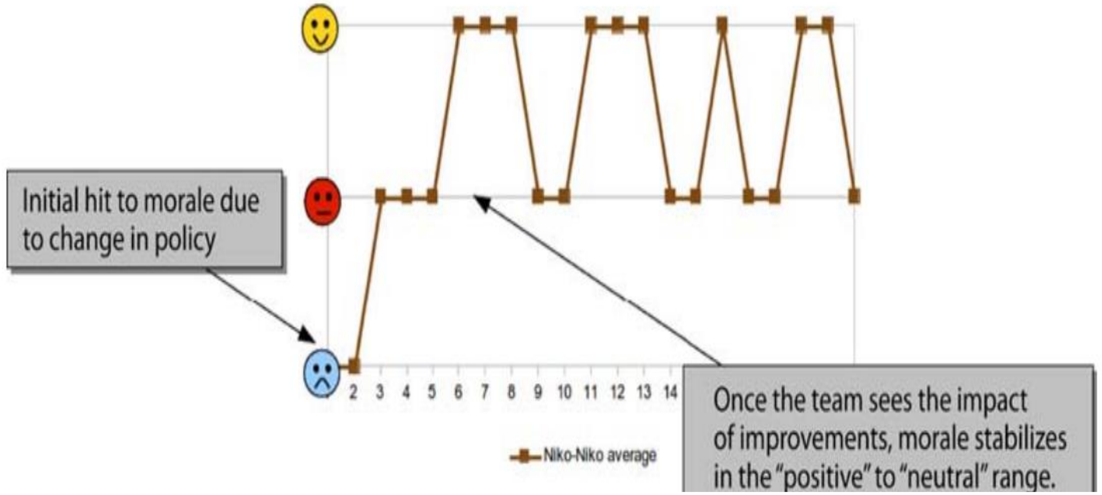
- *This pattern usually occurs when a team feels pressure to deliver as rapidly as possible (regardless of whether the pressure is real or perceived), and when they believe the best way to deliver quickly is to cut corners with respect to technical practices*
- Korektivne akcije
  - helping the team understand the business value of delivering clean code in a predictable and sustainable way
  - helping the team learn (if necessary) and adopt generally accepted good software development practices

Korektivne akcije (2)


Velocity stabilizes.


Test coverage improves to an acceptable level.


Code complexity stabilizes.


Initial hit to morale due to change in policy

Once the team sees the impact of improvements, morale stabilizes in the "positive" to "neutral" range.

Šablon 2

- *Velocity looks good, but little is delivered*
- Posmatra se time-box iterativni model, zahtjev je da se postigne visoka vrijednost za velocity, međutim veliki je broj ograničavajućih faktora:
  - Struktura organizacije - *aligning value-producing activities with the value stream*
  - Nadzor – *at the end of the delivery chain by including review and approval steps before any software is permitted into the production environment*
  - Ograničenja u resursima - *To support incremental delivery, development teams must have control of the resources they need to deliver fully tested software in each iteration*
  - xaf

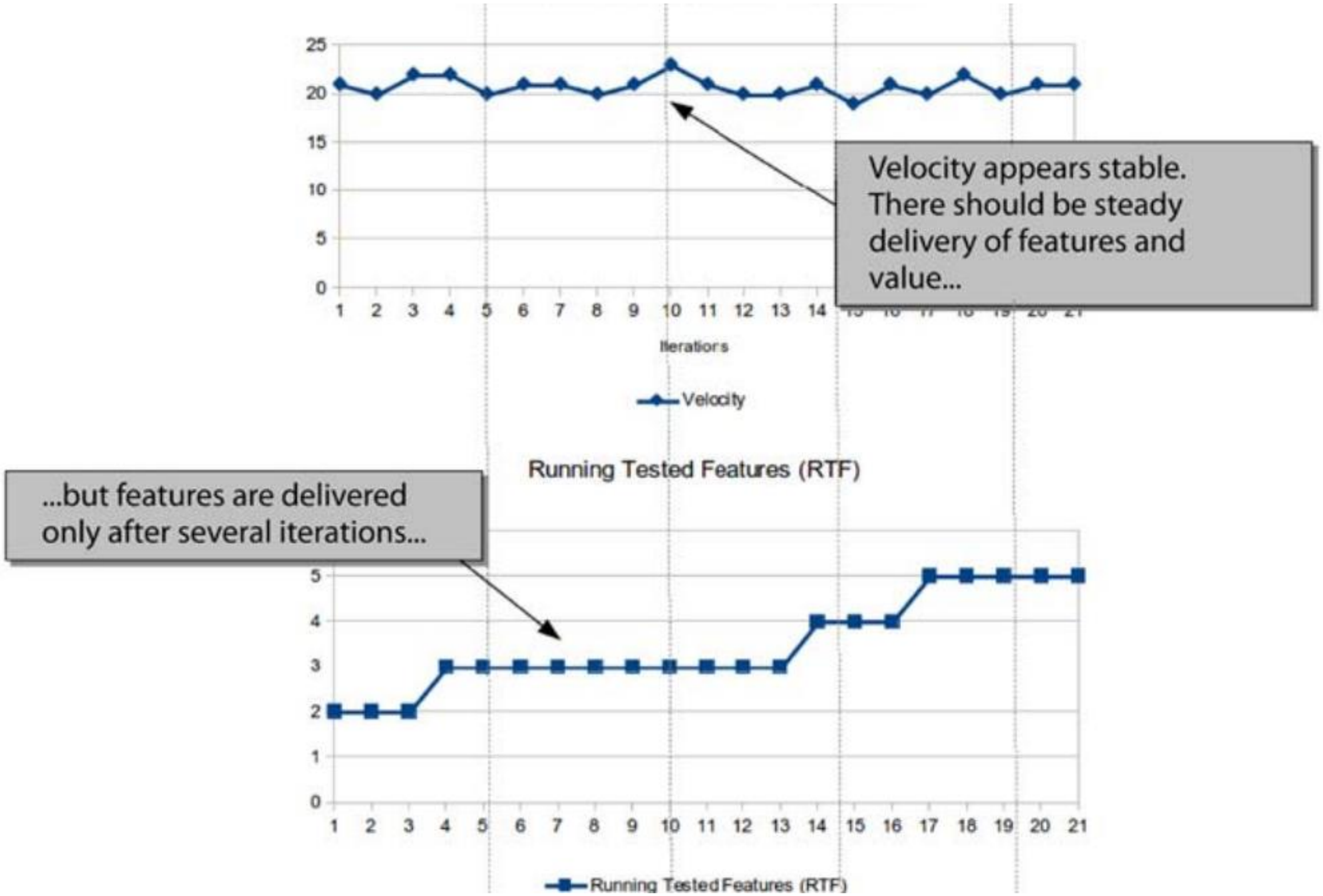
Šablon 2 - nastavak

- Ograničavajući faktori:
  - Manuelni delivery - *manual methods for building, packaging, testing, deploying, and monitoring the software they develop*
  - Zavisnost od third-party biblioteka
  - Iterativni waterfall model
  - *Functional silos - some individuals to be overloaded and others to be idle at any given time*
  - Scarce specialists
  - Meetings
  - *Time management - to function on an interrupt-driven basis*
  - „Skrivanje“ stvarnih performansi – *gaming velocity by manipulating their user-story sizing or estimation*

Gaming velocity

- Kombinacija metrika
  - Velocity
  - Running tested features - *The number of software features running in a production or staging environment with all automated tests passing*
  - Earned business value (EBV) - *The amount of relative business value delivered to date*
  - Throughput - *The number of features delivered to production per release*

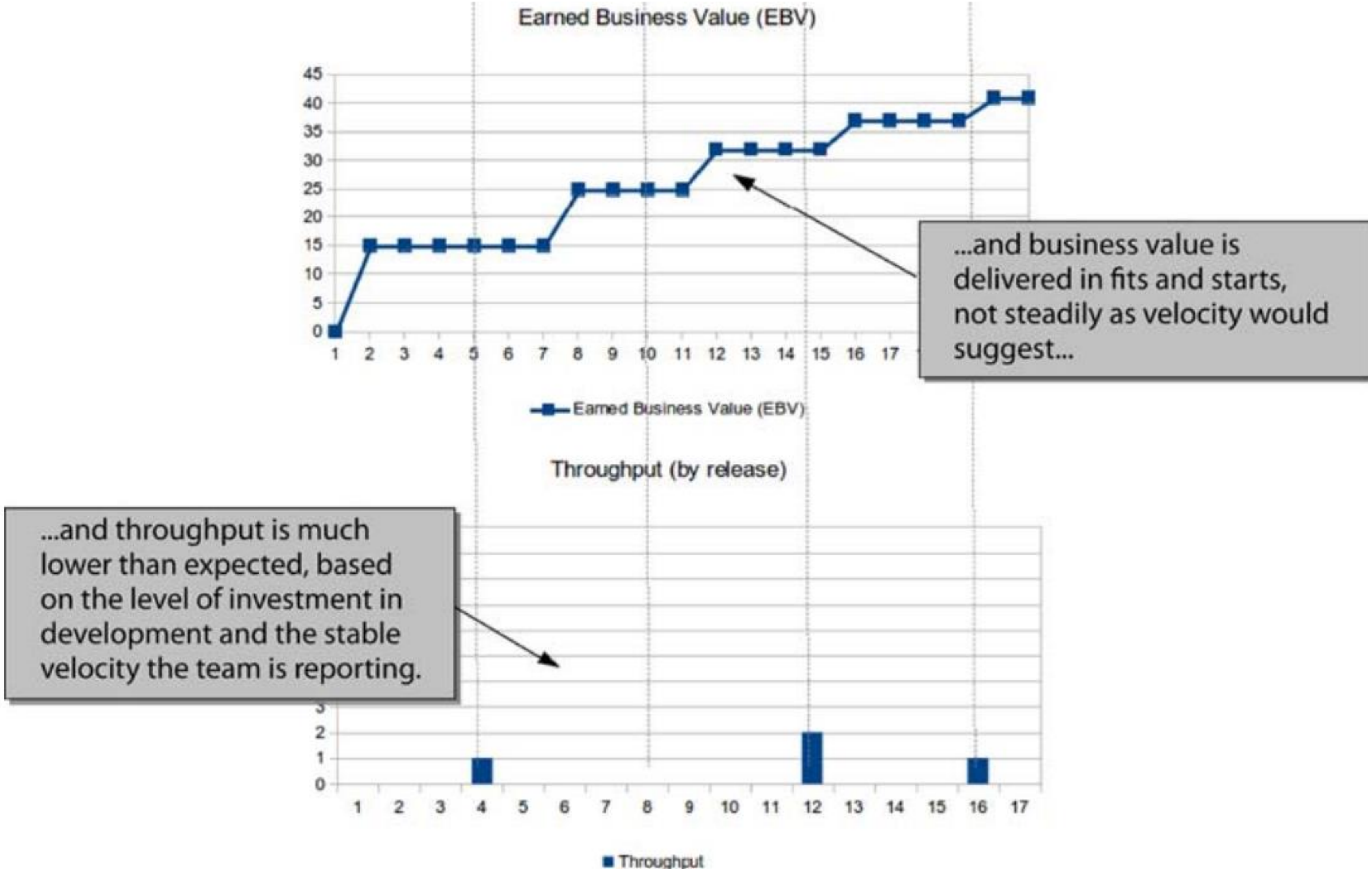
Gaming velocity (2)



Velocity appears stable. There should be steady delivery of features and value...

...but features are delivered only after several iterations...

Gaming velocity (3)



...and business value is delivered in fits and starts, not steadily as velocity would suggest...

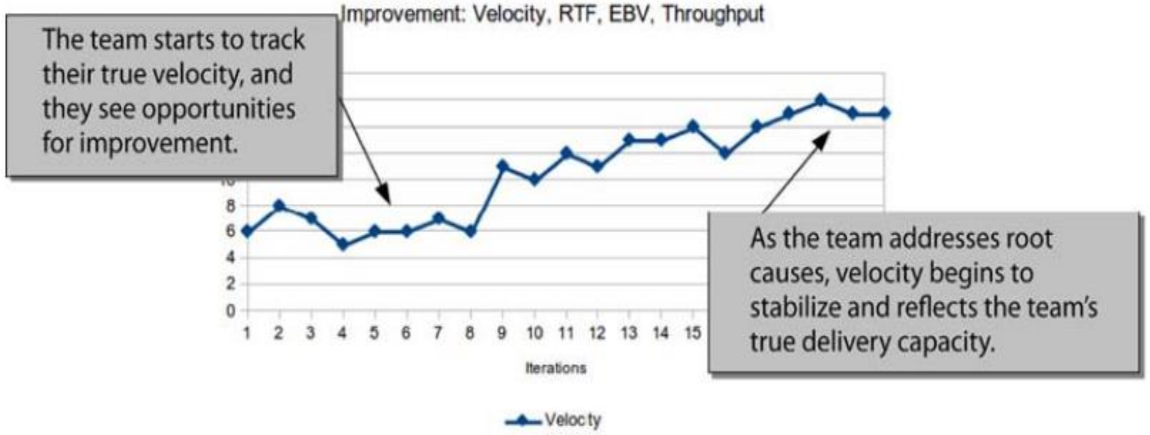
...and throughput is much lower than expected, based on the level of investment in development and the stable velocity the team is reporting.

Korektivne akcije

- Realno i iskreno dodjeljivanje story poena umjesto *gaming with velocity* da bi velocity grafikon izgledao kako treba

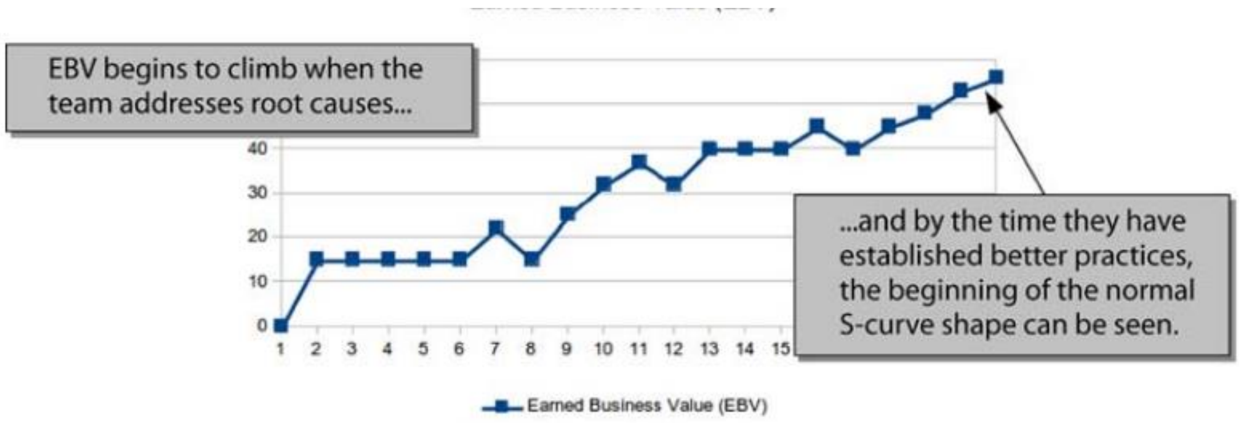
- Inicijalno ovo neće unaprijedi ostale mjere ali će moguće ukazati na root cause loših delivery performansi

Korektivne akcije (2)



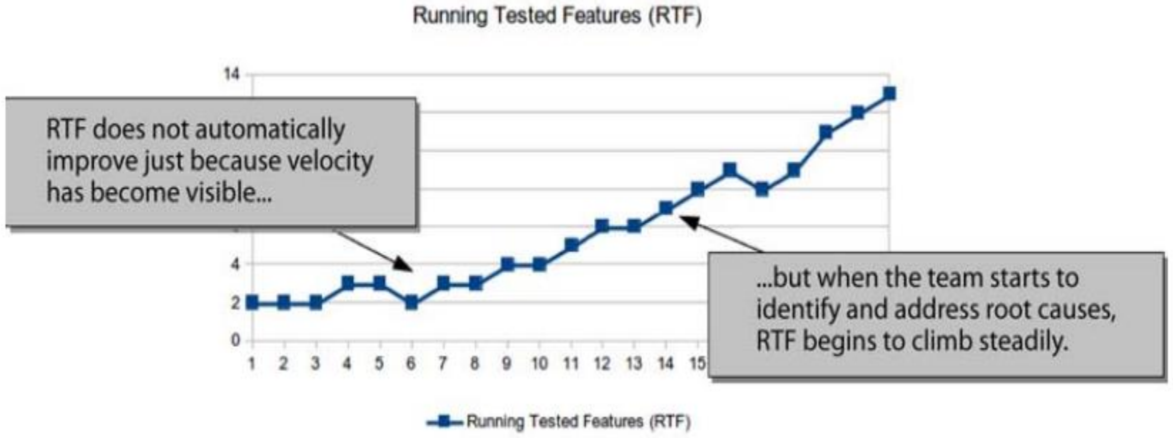
The team starts to track their true velocity, and they see opportunities for improvement.

As the team addresses root causes, velocity begins to stabilize and reflects the team's true delivery capacity.



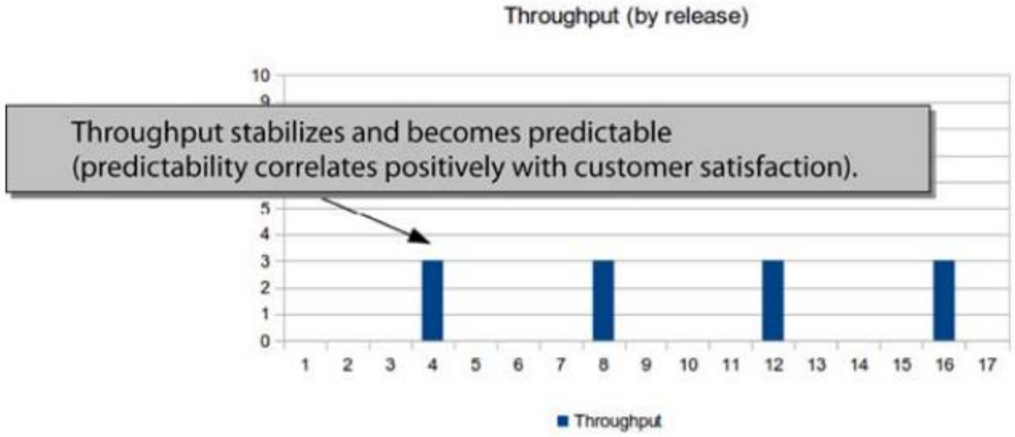
EBV begins to climb when the team addresses root causes...

...and by the time they have established better practices, the beginning of the normal S-curve shape can be seen.



RTF does not automatically improve just because velocity has become visible...

...but when the team starts to identify and address root causes, RTF begins to climb steadily.



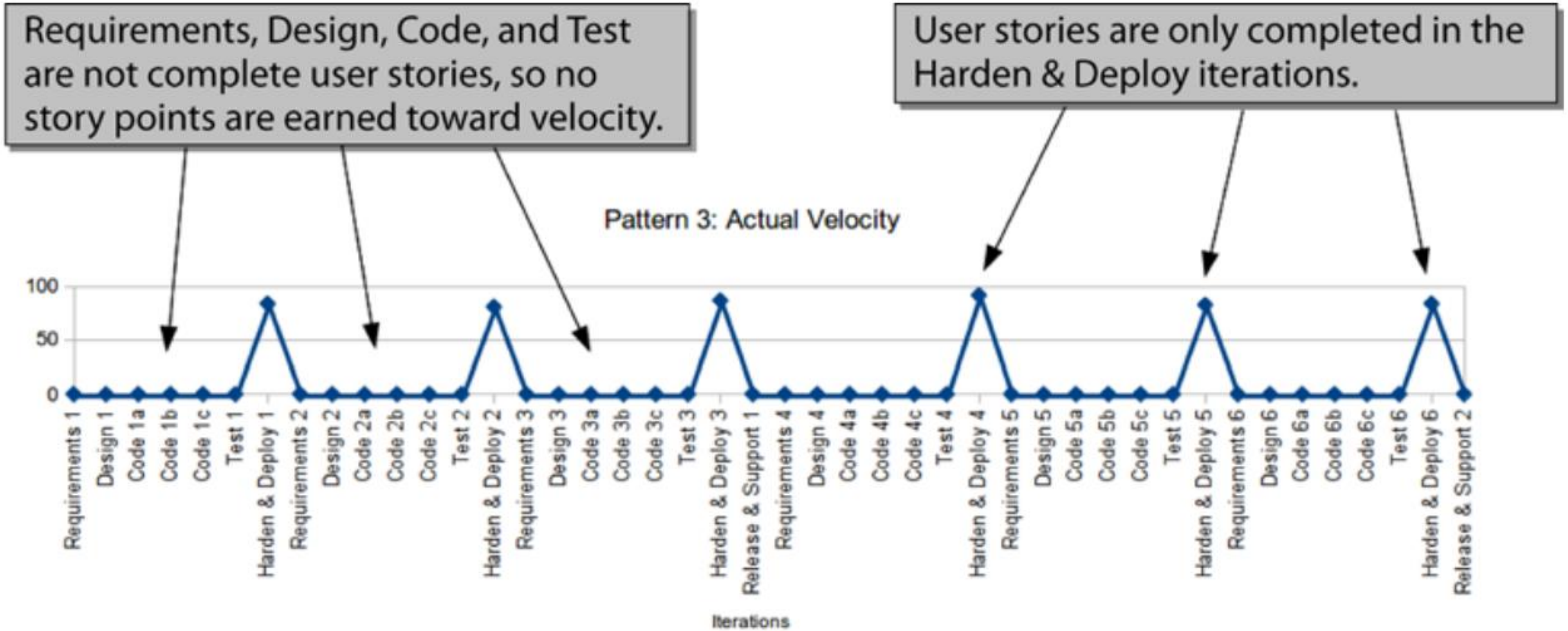
Throughput stabilizes and becomes predictable (predictability correlates positively with customer satisfaction).

Šablon 3

- *Linear workflow packaged in time-boxed iterations*
- Iterativni pristup ne postavlja uslov u smislu uniformne dužine iteracija i definisanja funkcionalnosti koje moraju biti završene tokom jedne iteracije, dovoljno je da *something usable is delivered in each iteration*
- Time-box iteracija: *production-ready solution increments to be delivered in each iteration, and for the iterations to be the same length*
- Linearni workflow podijeljen na time-box iteracije: *a Requirements iteration, a Design iteration, three Code iterations, a Test iteration, and a Harden & Deploy iteration*

Šablon 3 - nastavak

- Velocity broji samo kompletirane work iteme pa će prvi story poeni biti „zarađeni“ tek u posljednjoj fazi

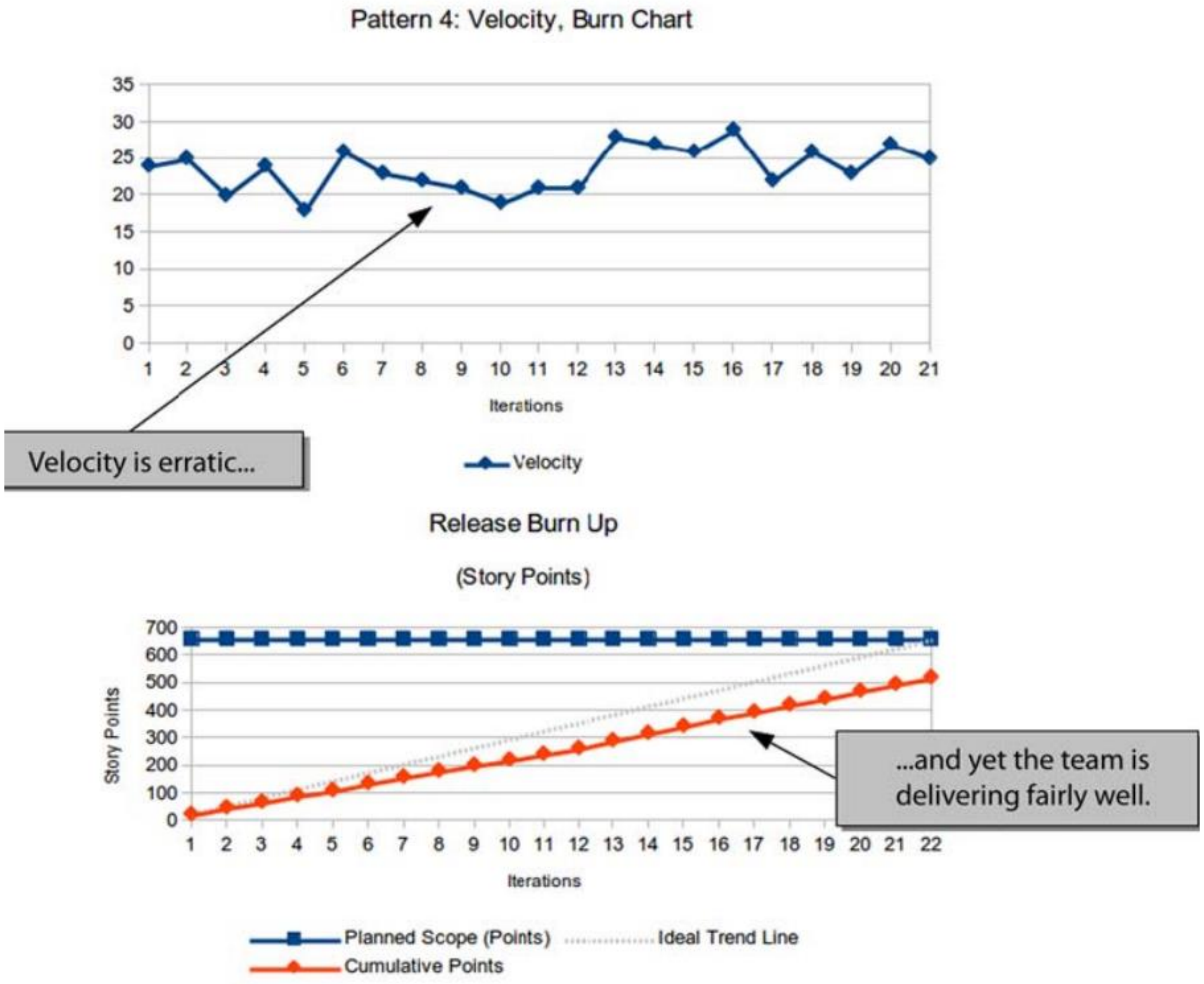


Requirements, Design, Code, and Test are not complete user stories, so no story points are earned toward velocity.

User stories are only completed in the Harden & Deploy iterations.

Šablon 4

- *Erratic velocity but stable delivery*



Pattern 4: Velocity, Burn Chart

Velocity is erratic...

Release Burn Up

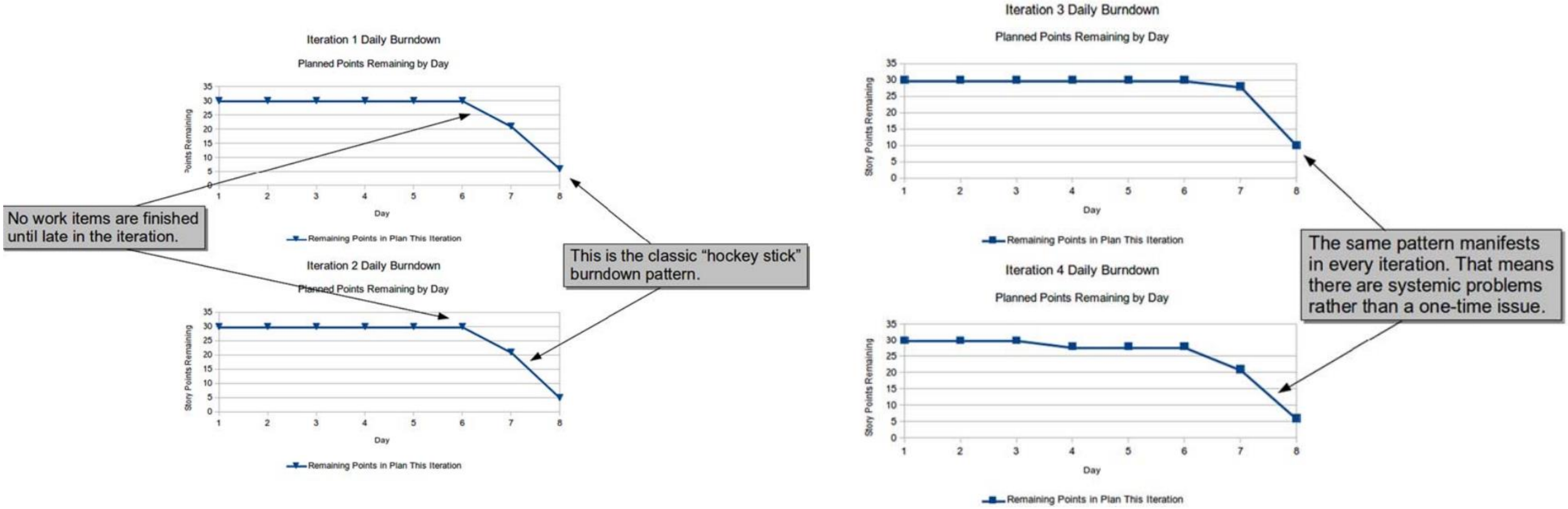
(Story Points)

...and yet the team is delivering fairly well.

Šablon 4 - nastavak

- Varijacije u velocity grafikonu su upozorenje!
  - *Is the team using technical practices that cause technical debt to accumulate in the code base?*
  - *Is the team overloaded with work?*
  - *Is the team’s morale in decline?*
  - *Are there external dependencies that cause the team to put work items on hold while they wait for turnaround from other teams or external suppliers?*

Šablon 4 - nastavak



No work items are finished until late in the iteration.

This is the classic “hockey stick” burndown pattern.

The same pattern manifests in every iteration. That means there are systemic problems rather than a one-time issue.