

Jak zacząć pracę z
metrykami?

Cele warsztatu

Znaleźć odpowiedzi na pytania:

- jak zaangażować zespół w pracę z metrykami,
- jak przełożyć teorię metryk do pierwszego użycia w zespole,
- jak wykorzystywać metryki na Retrospektywie i po co.

12 zasad mierzenia

The Twelve Rules for Metrics



1. Measure for a purpose



2. Shrink the unknown



3. Seek to improve



4. Delight all stakeholders



5. Distrust all numbers



6. Set imprecise targets



7. Own your metrics



8. Don't connect metrics to rewards



9. Promote values and transparency



10. Visualize and humanize



11. Measure early and often



12. Try something else

1. Mierz w określonym celu

To, co zostanie zmierzone, zostanie zarządzane,
a to, co zostanie zarządzane, zostanie zrobione.

Musisz zawsze rozumieć, dlaczego mierzysz.
Metryki nie są celem samym w sobie. To tylko
środek do celu. Wszystko zaczyna się od
“dlaczego”.

2. Zmniejsz nieznane

Nie wszystko da się zmierzyć. Jak zmierzyć szczęście osoby lub pracownika?

To, co możesz zrobić, to zmierzyć inne małe wskaźniki i pośrednie wartości, które sumują się do lepszego zrozumienia szerszego obrazu.

3. Staraj się udoskonalać

Przy wszystkich dostępnych informacjach i liczbach (duże zbiory danych) łatwo jest wyglądać na zajętego.

Nie mierz tylko rzeczy, które sprawiają, że będziesz dobrze wyglądać. Skoncentruj się na tym, co umożliwia Ci lepszą pracę.

4. Zadowolaj wszystkich interesariuszy

Twoja praca zależy od innych,
a inni zależą od ciebie.

Nigdy nie optymalizuj tylko dla jednego interesariusza. Mierz swoją pracę z wielu perspektyw.

5. Nie ufaj wszystkim liczbom

Obserwatorzy zwykle wpływają na swoje własne metryki i są obciążeni z powodu różnego rodzaju stronniczości.

Miej zdrowy, sceptyczny stosunek do wszelkich zbieranych liczb.

6. Wyznacz nieprecyzyjne cele (kierunek)

Jeśli wyznacysz target dla innych, będą oni działać pod ten target, zamiast dążyć do pierwotnego celu.

Spraw, aby cele były ogólnikowe i bardziej przypominały kierunek. Stawiaj ambitne cele, których osiągnięcie nie jest całkowicie możliwe.

7. Bądź właścicielem swoich metryk

Każdy jest odpowiedzialny za swoją pracę, a metryki pomagają nam ulepszać tę pracę.

Dlatego każdy powinien być odpowiedzialny za własne metryki.

8. Nie łącz metryk z nagrodami

Gdy ludzie podążają za “marchewką na kiju”, aby zdobyć nagrodę, to tracą z oczu pierwotne cele i szerszy obraz.

9. Promuj wartości i transparentność

Wspólne wartości i transparentność mogą zmniejszyć chęć oszukania systemu.

Spraw, aby liczby, zasady, metryki i cel były transparentne dla wszystkich.

10. Wizualizuj i uczłowieczaj dane

Nie prezentuj tylko samych liczb,
na które ludzie nie będą patrzeć.

Spraw, aby dane wyglądały ciekawie, ładnie i
fajnie się z nim pracowało.

11. Mierz wcześniej i często

Mierz wystarczająco często, aby mieć pewność, że problemy nie urosną do zbyt dużych rozmiarów i nie staną się zbyt ryzykowne.

Prawdopodobnie mierz też częściej niż robisz to teraz.

12. Spróbuj czegoś innego

Od czasu do czasu zmieniaj swoje metryki, aby odkryć inne perspektywy i różne spojrzenia.

The Twelve Rules for Metrics



1. Measure for a purpose



2. Shrink the unknown



3. Seek to improve



4. Delight all stakeholders



5. Distrust all numbers



6. Set imprecise targets



7. Own your metrics



8. Don't connect metrics to rewards



9. Promote values and transparency



10. Visualize and humanize



11. Measure early and often



12. Try something else

Jakie problemy możemy zaadresować?

- zadowolenie z produktu → NPS
- szybkość reakcji na zmiany na rynku → LT
- stabilność produktu → $n(\text{bug})$
- optymalizacja kosztów produkcji → $t(\text{blocked})$
- pojemność Sprintu → Throughput
- termin dowiezienia → $CT(0.85)$

40+ Metrics for Software Teams

The following listing is intended as a starting point for conversation and discussion. Choose one or two that make sense for your team / organization and add them to your current dashboard. Then rinse and repeat over time.

People/Team: Human Elements

This group of metrics reveals issues that impact a team's sustainable place and level of engagement.

- › Team Happiness / Morale / Mood
- › Gallop Q12
- › Team / Manager / Organization NPS
- › Percentage of time w/o interruptions
- › Trust between Leadership and Team
- › Learning Log
- › Team Tenure
- › Phone-a-Friend Stats
- › Whole Team Contribution
- › Transparency (access to data, access to customers, sharing of learning, successes and failures)
- › Comparative Agility: Team mapping against the 12 agile principles (Geoff Watt's "Scrum Mastery")

Process Health Metrics

This category assess day-to-day delivery team activities and evaluates process changes.

- › Cumulative Flow Diagrams
- › Control Charts
- › Cycle Time
- › Percent Complete and Accurate
- › Time Blocked per Work Item
- › Story/Epic Lead Time
- › Successful Iteration Completion
- › Escaped Defect Resolution Time

Release Metrics

This group directs focus on identifying impediments to continuous delivery.

- › Escaped Defects
- › Release Success Rate
- › Release Time
- › Time Since Last Release
- › Cost Per Release
- › Release Net Promoter Score
- › Release Adoption / Install Rate

Product Development Metrics

These help measure alignment of product features to user needs.

- › Customer / Business Value Delivered
- › Risk Burndown
- › Value Stream Mapping
- › Sales Velocity
- › Product Forecast
- › Product Net Promoter Score (NPS)
- › User Analytics

Technical/Code Metrics

The following help determine quality of implementation and architecture.

- › Test Coverage
- › Unit/Regression Test Coverage
- › Build Time
- › Defect Density
- › Code Churn
- › Code Ownership
- › Code Complexity
- › Coding Standards Adherence
- › Crash Rate
- › Build Breaks
- › Technical Debt
- › Ratio of Fixing Work vs Feature Work

Andy Cleff

Andy is an experienced and pragmatic agile practitioner that takes teams beyond getting agile to embracing agile. His chief weapons are well asked questions, insightful retrospectives and an ability to withstand awkward silences. And if all else fails, beer.

✉ andycleff@icloud.com

🌐 andycleff.com

🌐 [linkedin.com/in/andycleff](https://www.linkedin.com/in/andycleff)

🐦 [@JustSitThere](https://twitter.com/JustSitThere)

🌐 agileuprising.com



Metryki procesowe - pierwszy krok dla SMa

1. Lead time
2. Cycle time
3. Throughput
4. Cumulative flow

Cycle time - pytania

- Czy mamy jakieś obserwacje odstające?
- Co wpłynęło na to, że powstała odstająca obserwacja? Czy znamy przyczyny? Czy można było tego uniknąć?
 - Czy zadanie czekało w jakimś stadium czy wymagało tak dużo pracy? Dlaczego czekało?
 - Co możemy zrobić w przyszłości inaczej, by skrócić nasz CT?
- Przy 85% percentylu dostarczamy zadania do X dni roboczych:
 - Czy to jest dla nas ok, że dostarczamy w tyle dni? Jakie ma to implikacje?
 - Co możemy zrobić, by zadania były mniejsze i mieściły się w Sprincie?
 - Czy wiecie, że system jest bardziej wydajny, więcej jesteśmy w stanie dostarczyć użytkownikom, gdy zadania są mniejsze, bo płynniej się nad nimi działa? Co możemy w związku z tym zmienić w naszym systemie?
 - Czy limit WiP na którejś kolumnie mógłby pomóc?
- Warto dążyć do tego, by zadania były realizowane w podobnym okresie czasu, ponieważ buduje to naszą przewidywalność. Co możemy w związku z tym zrobić?
- Dodatkowo jak dążymy do jak najmniejszych zadań to równocześnie dopracowujemy przepustowość systemu, bo im mniejsze kawałki pracy tym idzie to sprawniej.

Throughput - pytania

- Ile zadań kończymy w danej jednostce czasu?
- Jak stabilny jest nasz system pracy?
- Dlaczego liczba ukończonych zadań wygląda w taki sposób? Co na to wpływa?
- Jakie widzimy trendy lub wzorce?
- Co możemy powiedzieć o tych trendach lub wzorcach?
- Co się zadziało, że nagle T. poszedł mocno w górę lub nagle spadł mocno w dół?
- Czy sposób dzielenia zadań się zmienił/miał jakiś wpływ?
- Czy podzielenie zadań spowodowałoby dostarczenie czegoś więcej użytkownikom?
- Ile zadań warto brać do kolejnego Sprintu?
- Czy czegoś da się uniknąć w przyszłości? (np. niedostarczenia niczego)
- Jak możemy ulepszyć przepustowość naszego systemu pracy?

Status na koniec sprintu - pytania

- Czy mamy elementy w "to do"? Czy nie wzięliśmy czegoś niepotrzebnie do Sprintu?
- Co się stało, że mieliśmy w Sprincie coś, co trzeba było cancelować?
- Dlaczego w danym okresie czasu mamy na koniec Sprintu rozpoczęte i niezakończone taski?
- Gdzie nasz system się blokuje? Co możemy z tym zrobić?
- Czy umiemy tak działać i planować, by nie otwierać nowych zadań na koniec Sprintu?
- Co się dzieje, że kończymy z czymś w ready for test?
- Co się dzieje, że coś jest w Tested i nie wyszło na produkcję na koniec Sprintu?

Jak jeszcze mocniej zaangażować zespół?

Angażujemy się w to na czym nam zależy, coś co jest dla nas ważne, na co mamy realny wpływ.

Jakie kroki wykonać?

1. Co wkurza zespół? Czego nie lubią?
 2. Zapiszcie objawy
 3. Zdefiniujcie problem
 4. Postawcie sobie cel
 5. Dobierzcie metryki
 6. Wymyślcie pomysły
7. Wdrażajcie i obserwujcie metryki

DZIĘKUJĘ

Justyna Rędzikowska

<https://www.linkedin.com/in/justyna-redzikowska/>