Jeśli jesteś nowicjuszem w programowaniu lub biznesie pracującym z software house, możesz wyobrazić sobie proces tworzenia oprogramowania jako zadanie od początku do końca. Klient przekazuje programistom instrukcje dotyczące produktu końcowego, a oni po prostu go tworzą. Gdy jest on gotowy, klient otrzymuje go i płaci za usługę. Jest to idea stojąca za rozwojem większości produktów. Jednak w przypadku oprogramowania cały proces jest bardziej wydajny. Jak jakikolwiek proces może być bardziej wydajny niż powyższy? Warto zatem przybliżyć nieco, jak wygląda tworzenie oprogramowania. Dziś przedstawię najlepszą metodologię, jaka kiedykolwiek zagościła na Ziemi. Może to być subiektywna opinia, ale możesz wyrobić sobie własny osąd na podstawie poniższych faktów. Żaden z poniższych faktów nie jest kłamstwem!

Kluczowe założenie na początek

Ten artykuł z definicji nie obejmuje każdego szczegółu każdego tematu , który jest w nim wspomniany. Jest to wprowadzenie do Agile jako koncepcji i wprowadzenie do niektórych zwinnych frameworków, które mogą być przydatne w tworzeniu oprogramowania (Scrum, Kan-Ban & ScrumBan). Istnieją jednak inne artykuły na tym blogu oznaczone tagiem „Agile for busniess”, które analizują perspektywy tego tematu.

Na koniec – prosimy o dokładne zapoznanie się z ostatnią sekcją tego dokumentu (“Przydatne źródła wiedzy – książki/eseje/strony internetowe/kanały YT itp. “), gdzie można znaleźć wiele źródeł wiedzy. Niektóre z nich (książki) znajdują się w bibliotece Sailing Byte’ reszta jest dostępna całkowicie za darmo w Internecie!

>

Dlaczego Sailing Byte stara się uczyć, wdrażać (w razie potrzeby) i dzielić się wiedzą na temat Agile?

Sailing Byte jest w ciągłym procesie doskonalenia Scrum/Scrumban. Uznaliśmy, że istnieją potencjalne korzyści organizacyjne z wprowadzenia zwinnych ram do naszych bieżących projektów rozwoju oprogramowania.

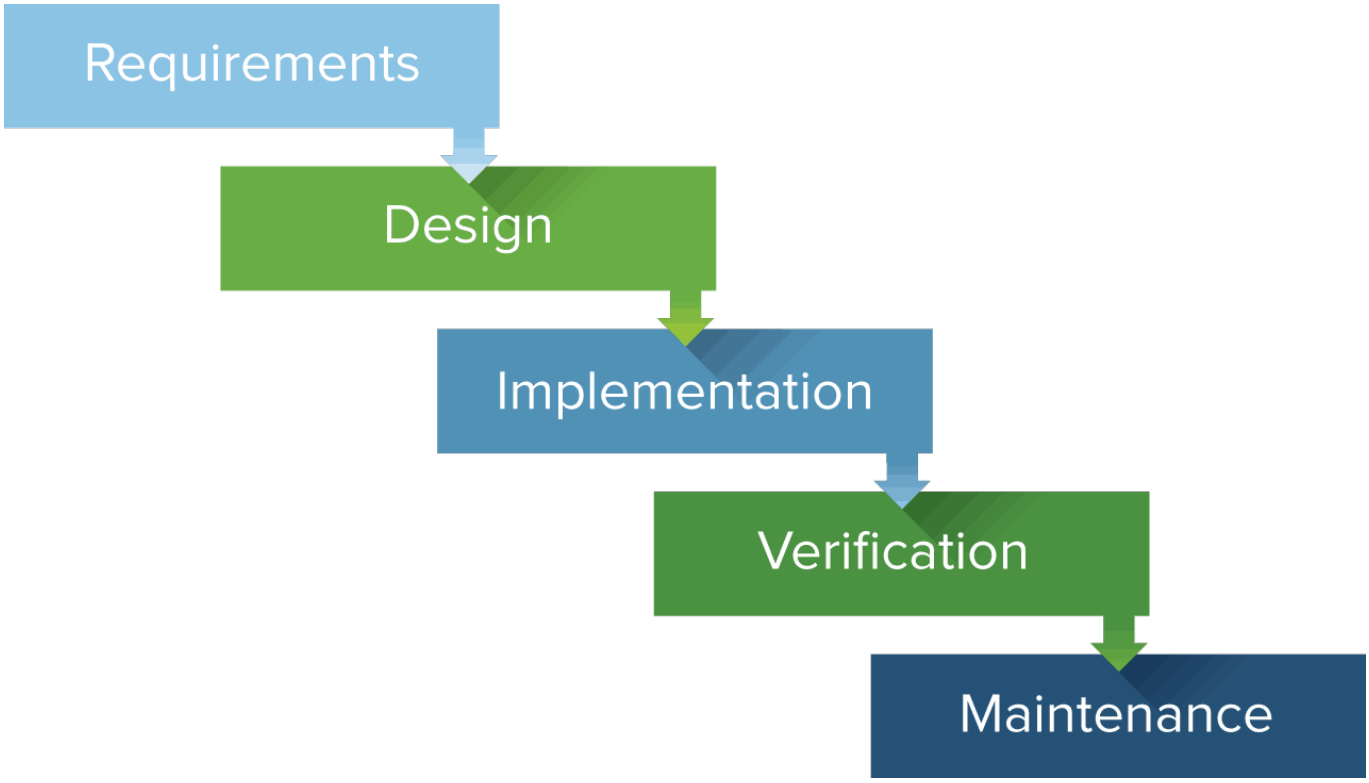
W ostatecznym rozrachunku chcemy dostarczać wysokiej jakości kod (a tym samym większą wartość dla klientów końcowych) w bardziej przewidywalnym tempie w ramach całego portfela projektów. Aby osiągnąć ten cel, być może będziemy musieli zmienić sposób pracy i współpracy z niektórymi interesariuszami. Wciąż uczymy się, jak głęboko możemy zastosować te zmiany w różnych projektach.

Podzielenie się wiedzą na temat metodyk Agile zarówno z obecnymi, jak i przyszłymi klientami Sailing Byte przynosi same korzyści. Jest to oczywista sytuacja win-win, gdy obie strony od samego początku rozumieją, jak wygląda cały proces i jakie są korzyści z określonego podejścia.

Co oznacza „Agile” w kontekście tworzenia oprogramowania?

Termin zwinny oznacza być *“zdolnym do szybkiego i łatwego poruszania się.”* Świat rozwoju oprogramowania z lat 80’ i 90’ był daleki od możliwości szybkiego i łatwego zarządzania wszelkimi zmianami. Każda potrzeba biznesowa nowego oprogramowania musiała podążać ścisłą “wodospadową” ścieżką rozwoju

>



Nie ma nic złego z definicji w tym podejściu (Waterfall). Rozsądnie jest pracować nad wszystkimi krokami (wymagania, projekt, implementacja, weryfikacja, utrzymanie) jeden po drugim. Na przykład, gdy zostaniesz poproszony o zbudowanie mostu, nie byłoby nic złego w pierwszym zebraniu wymagań (kto potrzebuje tego mostu i do czego), następnie zaprojektowaniu go (sporządzenie ścisłego planu technicznego), a następnie wdrożeniu tego planu dokładnie zgodnie z planem (rozpoczęcie prac budowlanych). Na koniec trzeba będzie zająć się weryfikacją (sprawdzić, czy most jest bezpieczny i służy swojemu celowi) i konserwacją (upewnić się, że most się nie zawali i będzie użyteczny dla klientów końcowych przez następne 100 lat).

Most musi być zbudowany ze **stałym zestawem wymagań** aby **zaspokajać jego potrzeby** co może nastąpić tylko wtedy, gdy ostatecznie **zbudujesz cały most. Nie ma wartości dla klienta końcowego w moście wykonanym w 65% lub nawet 99,9% Aby szybciej dostarczać wartość nie można zmieniać planów w trakcie prac budowlanych. Jest to raczej oczywiste.**

Jeśli zastosujesz tę metodę (kaskadową) do rozwoju oprogramowania, stanie się

jasne, że klient końcowy będzie musiał czekać na **wartość** aż do samego końca całego procesu. Będziesz musiał znać **wszystkie wymagania** na samym początku i **nie będziesz mógł ich zmienić** bez gotowości do poradzenia sobie ze wszystkimi ryzykami, które wiążą się z możliwym 2-3 razy dłuższym okresem rozwoju. (marnotrawstwo budżetu, marnotrawstwo zdolności produkcyjnych firmy, presja ze strony interesariuszy itp.)

Zadajmy sobie teraz pytanie, czy świat zmienia się teraz w 2021 roku szybciej niż zmieniał się w latach 80-90? Czy klienci zmieniają swoje zachowania i częściej wybierają inne aplikacje do rozwiązywania swoich problemów/potrzeb? Czy produkty muszą zmieniać się częściej ze względu na to “przyspieszenie świata” ?

Odpowiedzi na te pytania są raczej oczywiste, ale wyobraźmy sobie, że istnieje odważna grupa inżynierów oprogramowania, którzy widzą, że świat “przyspiesza” znacznie szybciej niż inni. Ci amerykańscy i angielscy inżynierowie oprogramowania zaproponowali w lutym 2001 roku “Manifesto for Agile Software Development” (później “Agile Manifesto”) – łysą deklarację, która zmieniła przyszłość rozwoju oprogramowania.

Manifest Zwinnego Tworzenia Oprogramowania – co zawiera?

Odkrywamy lepsze sposoby tworzenia oprogramowania, robiąc to i pomagając innym to robić. Poprzez tę pracę doszliśmy do wartości:

- **Jednostki i interakcje** nad procesami i narzędziami
- **Pracujące oprogramowanie** nad kompleksową dokumentacją
- **Współpraca z klientem** negocjowanie umów
- **Reagowanie na zmiany** nad przestrzeganiem planu

To znaczy, podczas gdy istnieje wartość w przedmiotach po prawej stronie, bardziej cenimy przedmioty po lewej stronie.

Ponadto grupa inżynierów zaproponowała listę 12 **zasad stojących za Manifestem Agile.**

W 2001 roku grupa programistów, którzy od jakiegoś czasu stosowali nowe podejście, opracowała Manifest Agile. Zestaw zasad dokumentował ich wspólne

przekonania i doświadczenia dotyczące tego, jak powinien wyglądać nowoczesny proces tworzenia oprogramowania. Manifest Agile został podpisany przez Kena Schwabera, Kenta Becka, Martina Fowlera, Rona Jeffriesa i Jeffa Sutherlanda. Istnieje on do dziś na oficjalnej stronie internetowej podejścia Agile.

.

Agile wierzy w znaczenie współpracy, samoorganizacji i zdolności do dostosowywania się do ciągłych zmian. Przeciwstawia się nadmiernie skomplikowanemu procesowi dokumentacji, sztywnemu zarządzaniu i surowym zasadom. Ludzie pracują z myślą o głównym celu – szybkim stworzeniu produktu wysokiej jakości.

>

Przestrzegamy następujących zasad:

1. Naszym najwyższym priorytetem jest zadowolenie klienta poprzez wczesne i ciągłe dostarczanie wartościowego oprogramowania.
2. Przyjmujemy zmieniające się wymagania, nawet na późnym etapie rozwoju. Zwinne procesy wykorzystują zmiany dla przewagi konkurencyjnej klienta.
3. Dostarczaj działające oprogramowanie często, od kilku tygodni do kilku miesięcy, preferując krótsze terminy.
4. Biznesmeni i programiści muszą współpracować codziennie przez cały czas trwania projektu.
5. Buduj projekty wokół zmotywowanych osób. Zapewnij im środowisko i wsparcie, którego potrzebują, i zaufaj im, że wykonają swoją pracę.
6. Najbardziej wydajną i skuteczną metodą przekazywania informacji zespołowi programistów i w jego obrębie jest rozmowa twarzą w twarz.
7. Podstawową miarą postępu jest działające oprogramowanie.Procesy zwinne promują zrównoważony rozwój.

   .

8. Sponsorzy, deweloperzy i użytkownicy powinni być w stanie utrzymać stałe tempo w nieskończoność.

9. Ciągła dbałość o doskonałość techniczną i dobry projekt zwiększa zwinność.
10. Prostota – sztuka maksymalizacji ilości niewykonanej pracy – jest niezbędna.
11. Najlepsze architektury, wymagania i projekty wyłaniają się z samoorganizujących się zespołów.
12. W regularnych odstępach czasu zespół zastanawia się nad tym, jak stać się bardziej efektywnym, a następnie dostraja i odpowiednio dostosowuje swoje zachowanie.

więcej o Agile Manifesto:
Agilemanifesto.org

Powstanie takiego manifestu i jego zasad miało ogromny wpływ na świat rozwoju oprogramowania. Wewnątrz branżowych “największych mózgów” pojawiła się chęć pozbycia się “wodospadu” jako **jedynego sposobu** na wytwarzanie oprogramowania.

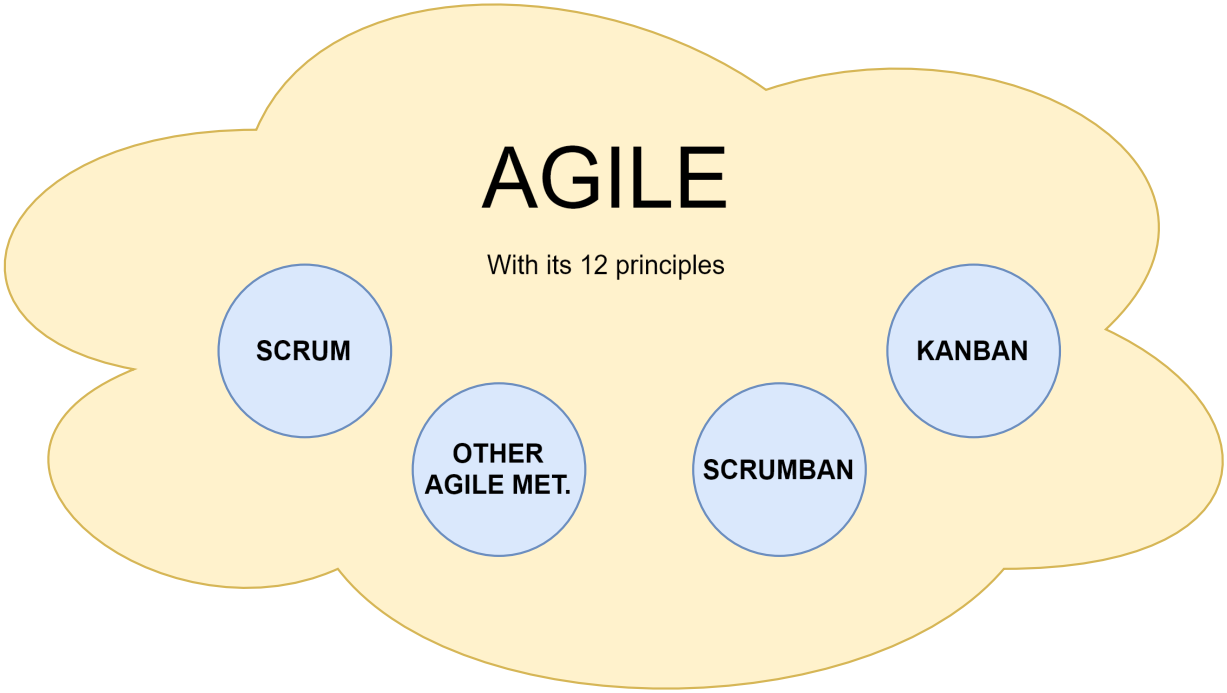
>

Podejście, które zaproponowali, daje główną przewagę w postaci szybszego **dostarczania wartości** klientowi końcowemu w połączeniu z szybszą **informacją zwrotną.** W rezultacie jesteśmy teraz w stanie obniżyć ryzyko związane z dostarczeniem projektu, który pod koniec procesu rozwoju **nie zaspokoi potrzeb klienta końcowego.** Potrzeby te mogą zmienić się w każdej chwili i będą zmieniać się jeszcze częściej w XXI wieku.

Czym są metodyki zwinne?

Zacznijmy od podstaw. Zwinne metodyki to podejścia do zarządzania tworzeniem oprogramowania. Istnieje kilka z nich, ale ich główną cechą jest istnienie ogólnych założeń projektowych, które są stopniowo dostosowywane na etapach rozwoju. Korekty te pochodzą od klienta, który dokonuje ponownej oceny swoich wymagań w oparciu o efekty prac.

>



Scrum vs Agile – na czym polega różnica?

Mówiąc o Agile, mogłeś natknąć się na termin Scrum. Podczas gdy Kanban i Scrumban nie są zbyt często mylone z Agile, z jakiegoś powodu Scrum jest czasami używany jako synonim Agile. Ale to nie to samo. Jaka jest więc definicja Scrum? Jest to jedna z ram (sposobów postępowania, narzędzi), które można wykorzystać w metodologii Agile. Tak więc Agile to ruch, ogólna idea, a Scrum to jedna z metod realizacji celów Agile. Wiele osób traktuje Agile i Scrum jako różne podejścia. Jednak koncepcja Scrum vs Agile nie powinna istnieć. Scrum wywodzi się z Agile, a oba te podejścia mają wspólny cel, którym jest szybki i wysokiej jakości rozwój produktu.

>

Co to jest Scrum?

Znając definicję Scrum, możemy teraz skupić się na znaczeniu Scrum. Scrum dzieli projekt na małe części. Są to fazy, podczas których zespół rozwija określone funkcje. Etapy te nazywane są Sprintami i trwają około czterech tygodni. Zmiany

wprowadzane podczas Sprintów powinny być wartościowymi ulepszeniami widocznymi dla użytkowników oprogramowania. Istnieją inne interesujące aspekty tego podejścia, takie jak:

>

- rola Właściciela Produktu polegająca na zebraniu wstępnych wymagań potrzebnych do rozpoczęcia sprintu i kontrolowaniu ich przestrzegania
- planowanie sprintu – wybór celu każdej fazy i zadań o najwyższym priorytecie, nad którymi należy pracować w pierwszej kolejności
- prowadzenie krótkich codziennych spotkań (Daily Scrums), na których zespół omawia pracę wykonaną poprzedniego dnia oraz pracę do wykonania w dniu spotkania
- Przegląd Sprintu zaplanowany po każdym sprincie podsumowuje pracę, kończy ją i wyznacza grunt pod ulepszenia.

Jak powstał Scrum?

Znaczenie Scrum zostało wprowadzone w 1986 roku przez Hirotakę Takeuchi i Ikujiro Nonakę w artykule w Harvard Business Review. Podkreślili oni dwie cechy Scrum (elastyczność i szybkość) i porównali nakładające się na siebie fazy rozwoju produktu do gry w rugby. Podkreślili znaczenie pracy zespołowej i dążenia do osiągania lepszych wyników. Jednak dopiero po pewnym czasie nastąpiło wdrożenie znaczenia Scrum w tworzeniu oprogramowania. Ken Schwaber i Mike Beedle opisali metodologię Scrum w „Agile Software Development with Scrum” w 2001 roku. Schwaber założył również Scrum Alliance (2002), Scrum.org (2009) i opublikował The Scrum Guide, który definiuje framework Scrum. Dokument ten jest aktualizowany w razie potrzeby, zgodnie z celem Scrum – dążeniem do osiągania lepszych wyników.

>

Co to jest Ag Ag Ag?

Co to jest Agile?

Agile, jak wspomniano we wstępie, to zbiór metod opartych na dostosowywaniu się do stale zmieniających się wymagań

klienta. Stosuje się ją w celu szybkiego opracowania produktu najwyższej jakości. Kluczową częścią metodologii Agile jest struktura zespołów, które pracują nad projektem. Zazwyczaj brak jest jakiejkolwiek hierarchii organizacyjnej. Zespół jest wielofunkcyjny i samodzielnie zarządzany. Członkowie mogą decydować, w jaki sposób wyznaczone cele zostaną osiągnięte. Komunikacja między członkami jest kluczowa dla powodzenia projektu.

Jak zaczęło się Agile?

Agile został zaproponowany jako zmiana metodologii kaskadowej. Rozwój aplikacji internetowych był momentem, w którym programiści zaczęli poszukiwać bardziej wydajnej metodologii wprowadzającej iteracyjne i oparte na współpracy procesy. Wszystko to było spowodowane zwiększonym obciążeniem pracą i zróżnicowanym zapleczem wielu małych zespołów, które pracowały nad wymagającymi projektami. Najlepszą częścią Agile jest to, że przyszedł naturalnie. Innymi słowy, nikt nie usiadł pewnego dnia i nie powiedział: *„Podejście kaskadowe nie jest już dobre do tworzenia oprogramowania. Od teraz programiści będą pracować zgodnie z następującymi zasadami…”.* Daleko od tego! Agile rozwinęło się w wyniku pracy programistów. To właśnie ludzie tworzący oprogramowanie zapoczątkowali nowy sposób postępowania, tylko dlatego, że pozwolił im on lepiej pracować. To jest właśnie esencja rozwoju Agile – używać metod, które pomagają osiągnąć lepszy rezultat w najszybszym możliwym czasie. Czy to oznacza, że w Agile nie ma żadnych zasad? Wcale nie. Zasady istnieją. Po prostu nie są ściśle narzucone, ale przychodzą naturalnie. Ten zestaw zasad nazywany jest Manifestem Agile.

>

Dlaczego metodyki Agile są lepsze od poprzedniego podejścia?

Prostą odpowiedzią, która jest również zgodna z pierwszą zasadą Manifestu Agile, jest to, że podejście Agile jest po prostu szybsze i daje lepsze wyniki. Dzięki Agile programiści nadają priorytet temu, co powinno być priorytetem – produktowi końcowemu. Mniej przejmują się tym, w jaki sposób go rozwijają, dopóki wymagania są spełnione. Wiedzą, że dokumentacja i zasady powinny pomagać ludziom w pracy. Jednak zbyt wiele z powyższych spowoduje odwrócenie uwagi od istoty tej pracy. Ponadto podzielenie pracy na możliwe do zarządzania części (Sprinty) umożliwia programistom szybsze jej ukończenie. Przedstawiają oni wynik Sprintu klientowi i otrzymują nowe

wymagania. Jest to przeciwieństwo długiego, rozległego planu wszystkich etapów projektu, który może nawet okazać się bezowocny w miarę jego rozwoju.

Wreszcie, dzięki bardziej zrelaksowanemu i zorientowanemu na cel podejściu, deweloperzy są szczęśliwsi i lepiej zmotywowani. Kto nie byłby szczęśliwy, gdyby mógł pracować i osiągać sukcesy bez zbędnego zamieszania?

Nie mogę oprzeć się skojarzeniu Agile ze ścieżkami pożądania. W planowaniu urbanistycznym ludzie tworzą ścieżki pożądania, ponieważ są to skróty lub łatwiejsze trasy niż te utwardzone. Uważam, że jest to idealna metafora tego, dlaczego ludzie używają Agile w tworzeniu oprogramowania zamiast podejścia kaskadowego.

O frameworkach/narzędziach:

Przewodnik po Scrumie – wersja 2020
“Scrum z Kanbanem” przewodnik
scrum.org – wersja 2021
Labirynty Scruma – Jacek Wieczorek
Scrumban: Essays on Kanban Systems for Lean Software Development, Corey Ladas
Kan-Ban – Agile Project Management with Kanban, Eric Brechner
Succeeding with Agile: Software Development Using Scrum

Inspiracje :
scrumgroup.org
Scrum.org
Zainspiruj mnie Kuba – kanał PL YT
AgileRebels.org
Agile.org
Atlassian – kanał YT
Development that pays – kanał YT
kanbanblog.com

>

Książki luźno związane z tematyką Agile/organizacji/zarządzania:

Cargo Cult Programming – Richard P. Feynman calteches.library.caltech.edu/51/2/CargoCult.pdf
The Machine That Changed The World – James P. Womack, Daniel T. Jones, Daniel Roos
It’s Not Luck – Eliyahu M. Goldratt (TOC)
THE GOAL – Eliyahu M. Goldratt (TOC)
Slack – Tom DeMarco
Project Phoenix – Gene Kim & Kevin Behr, George Spafford
Elegant Puzzle – An Elegant Puzzle, Systems Of Engineering Management – WIlliam Larsson
Pięć dysfunkcji zespołu – Patric Lencioni
Extreme Ownership – Jocko Willink & Leif

SAILINGBYTE

Babin

Agile w codziennym życiu – Sailing Byte jako eksperci w podejściu

Dzisiejszy świat to zmiany, szybkie dostosowania i ulepszenia. Utknięcie w podejściu kaskadowym przyniosłoby efekt przeciwny do zamierzonego. Właśnie dlatego, gdy tylko opublikowano Manifest Agile, został on dobrze przyjęty przez wiele firm zajmujących się tworzeniem oprogramowania na całym świecie. Według zippla.com, 86% międzynarodowych twórców oprogramowania korzysta z Agile. Wiele znanych przedsiębiorstw, takich jak Apple, Google, IBM, Cisco, Microsoft i Spacex, wykorzystuje znaczenie Scrum i podejścia

Agile w swoich strukturach. Kto jeszcze go używa? Sailing Byte, oczywiście! Jeśli chcesz, aby Twój produkt został stworzony w jak najkrótszym czasie i zgodnie z najwyższymi standardami, Agile jest najlepszym rozwiązaniem. Umów się na rozmowę już dziś, aby omówić znaczenie Agile i Scrum w procesie tworzenia oprogramowania.

>