ScrumBut—Deviasi Implementasi Scrum di Sektor Industri (Studi Kasus: Ralali)

Muhammad Nadhif Fuadi[1]
Program Studi Informatika – Program Sarjana
Universitas Islam Indonesia
Yogyakarta, Indonesia
16523230@students.uii.ac.id

Teduh Dirgahayu[2]
Jurusan Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
teduh.dirgahayu@ uii.ac.id

***Abstract*—Scrum adalah kerangka kerja yang dirancang secara spesifik untuk memenuhi keinginan dan kebutuhan pelanggan yang berubah dengan cepat. *Scrum* mampu menangani produk kompleks yang kebutuhannya berubah dalam waktu singkat dengan mengedepankan kolaborasi tim, proses iterasi, dan peningkatan produk yang konsisten. Hanya saja, implementasi praktis di lapangan jarang mengikuti panduan *Scrum* yang ideal secara sempurna. Fenomena ini dikenal dengan istilah *ScrumBut* yaitu deviasi dari panduan *Scrum* yang dianjurkan dikarenakan berbagai alasan. Deviasi ini bisa dilandaskan oleh alasan yang valid dan pertimbangan yang matang tetapi juga mungkin dilakukan tanpa memahami kemungkinan konsekuensi yang ada. Untuk memahami konteks mengapa Ralali melakukan deviasi dalam implementasi *Scrum* mereka serta konsekuensi yang diterima dan solusinya, deviasi-deviasi yang ada dianalisis menggunakan *anti-patterns.* Dari hasil analisis didapatkan tiga *anti-patterns* yang menjelaskan konteks serta alasan terjadinya deviasi, model deviasi yang terjadi, solusi dari deviasi tersebut, waktu pengecualian di mana penggunaan deviasi dapat dibenarkan, dan kemungkinan konsekuensi yang ada dari penerapan deviasi tersebut dalam keseharian kerja.**

Keywords—Agile, Scrum, ScrumBut, Ralali, Anti-patterns

I. PENDAHULUAN

Ralali adalah perusahaan yang bergerak pada industri lokapasar dengan model bisnis *business to business* (B2B) atau secara spesifik adalah usaha mikro, kecil, dan menengah (UMKM). Ralali menyediakan jasa untuk pemenuhan kebutuhan mereka dari berbagai penyedia yang terpercaya dengan aman dan mudah.

Ralali memiliki konsep pasar daring vertikal: Business Innovation Group (BIG), di mana mereka mengelompokkan berbagai penyedia barang dan produk dari berbagai kategori bisnis secara spesifik. Konsep ini dibangun tujuan agar Ralali mampu mengenali kebutuhan unik setiap pelanggannya secara mendalam sehingga mampu memberikan solusi kepada UMKM agar kebutuhan bisnis mereka dapat terpenuhi. Beberapa kategori tersebut antara lain adalah produk usaha kuliner (BIG Resto), usaha perkantoran (BIG Office), usaha otomotif (BIG Auto), usaha kebutuhan ritel (BIG Mart), usaha bangunan (BIG Home), usaha *reseller* kebutuhan (BIG Mart), dan usaha jasa (BIG Agent).

Kebutuhan bisnis dan respons pelanggan yang dinamis, selalu berubah dalam jangka waktu singkat mengharuskan Ralali untuk cepat beradaptasi dalam proses pembangunan produknya. Atas dasar permasalahan tersebut, Ralali mempraktikkan *agile development* atau *agile methods.* Metode pengembangan perangkat lunak *Agile* adalah istilah untuk suatu kumpulan kerangka kerja dan praktik yang didasarkan pada nilai dan prinsip *agile manifesto* [1]. *Agile methods* ini kemudian dimanifestasikan dalam beberapa kerangka kerja, salah satunya adalah *Scrum* yang digunakan oleh Ralali dalam lingkungan kerja mereka.

Scrum adalah kerangka kerja yang dibangun untuk menangani produk yang kompleks dan berubah-ubah. Melalui proses iterasi dan peningkatan produk yang konsisten, *Scrum* memastikan produk yang potensial untuk digunakan selalu tersedia sehingga mampu memaksimalkan keuntungan dari masukan dan saran pelanggan dan mengontrol kemungkinan risiko yang ada [3]. Hal ini sejalan dengan kebutuhan Ralali di mana pola perilaku pelanggan dan kebutuhan bisnis mereka selalu berubah-rubah dalam waktu singkat. *Scrum* pada dasarnya sangat sederhana, terdiri dari hanya tiga peran, tiga artefak, dan lima kegiatan [4]. Yang pada tingkatan ini Ralali sudah mengimplementasikannya dengan baik.

Hanya saja, jika diteliti lebih mendetail ada beberapa aspek dalam lingkup kerja Ralali yang bergeser dari panduan *Scrum* yang ada. Praktik pergeseran atau deviasi *Scrum* dari panduan resmi pada implementasi di lapangan dikenal sebagai *ScrumBut.*

Deviasi-deviasi ini bisa saja berlandaskan alasan yang rasional dan melalui proses pertimbangan yang matang sehingga layak untuk dilakukan. Tetapi terlepas dari alasan yang melandasi deviasi tersebut, *Scrum* adalah kerangka kerja komprehensif yang berpijak pada proses empiris dan mengalami evolusi berulang-ulang dalam 20 tahun terakhir [6]. Karenanya suatu deviasi mempunyai risiko tinggi untuk mengaburkan nilai dan tujuan *Scrum* secara keseluruhan sehingga secara umum, praktik ini harus dihindari. Deviasi yang ada akan dipresentasikan secara sistematis menggunakan *anti-patterns.*

Tujuan dari makalah ini adalah untuk memberi gambaran bagi praktisi pengembangan perangkat lunak, atau bagi perusahaan yang baru ingin mulai mengimplementasikan *Scrum* pada kerja keseharian mereka bahwa terdapat deviasi yang terlihat sah dan aman untuk dilakukan tetapi sebenarnya sangat berbahaya.

II. DASAR TEORI

A. *Agile*

Pendekatan metode pengembangan perangkat lunak *Agile* dirumuskan pada tahun 2001 melalui *Agile Manifesto* yang merupakan hasil pertemuan 17 orang praktisi pengembangan perangkat lunak untuk menjawab kebutuhan industri akan metode pengembangan perangkat lunak yang cepat [3]. *Agile manifesto* adalah sebuah dokumen yang dibangun dengan 4 nilai dasar dan 12 prinsip [2].

Metode *Agile* adalah metode pengembangan yang menggunakan konsep iterasi untuk melakukan peningkatan produknya. Setiap iterasinya berbentuk kecil dan biasanya versi produk terbaru, dibuat dan diberikan kepada pelanggan setiap dua atau tiga minggu. *Agile* melibatkan pelanggan dalam proses pengembangan mereka agar mendapatkan umpan balik atau kritik dan saran dari para pelanggan. *Agile* menganggap proses desain dan implementasi sebagai inti dari

seluruh aktivitas yang ada pada proses pengembangan perangkat lunak sehingga mereka meminimalisasi proses dokumentasi dan lebih memilih jenis komunikasi informal dibanding pertemuan formal dan dokumentasi tertulis [3].

B. *Scrum*

Scrum dibangun untuk menjawab permasalahan komunikasi informal dan kurangnya dokumentasi tertulis pada *Agile*. *Scrum* menyediakan kerangka kerja untuk mengatur proyek *Agile* hingga pada tahap *top level* manajemen dan semua pihak yang terlibat dalam proyek tersebut mengetahui proses, kendala, dan kemajuan yang dialami proyek tersebut. Sampai pada tahap ini, *Scrum* adalah metodologi *Agile* karena *Scrum* menerapkan dan mengikuti nilai dan prinsip-prinsip yang ada pada *Agile Manifesto.* Akan tetapi *Scrum* hanya fokus menyediakan kerangka kerja bagi penyusunan dan manajemen proyek *Agile. Scrum* tidak mengharuskan penggunaan model pengembangan yang spesifik seperti *pair-programming* atau yang lainnya. Sehingga *Scrum* dapat dengan mudah diintegrasikan dengan proses yang sudah ada pada suatu perusahaan [3].

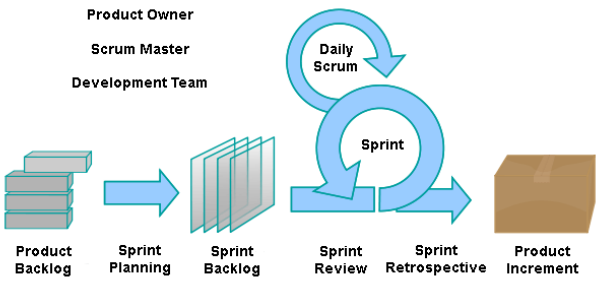

Fig. 1. Alur Scrum.

Seperti yang terlihat pada Fig. 1 *Scrum* terdiri dari tiga peran, tiga artefak dan lima kegiatan serta satu set peraturan yang mengikat keseluruhannya [4]. Setiap unsur yang ada pada *Scrum* mempunyai fungsi tersendiri yang bila disingkirkan tanpa memahami terlebih dahulu esensi dari unsur tersebut atau tanpa ada alasan yang jelas dan rasional, maka keseluruhan nilai dan keuntungan dari menggunakan *Scrum* itu sendiri terancam hilang. Meski memungkinkan untuk mengimplementasikan hanya beberapa bagian saja dari *Scrum*, hasil akhir yang didapat tidak bisa disebut *Scrum.* Suatu organisasi hanya bisa disebut telah berhasil mengimplementasikan *Scrum* ketika seluruh nilai dan fungsi *Scrum* itu sendiri ada dan digunakan [6].

Mayoritas perusahaan yang mengadopsi *Scrum* tidak menerapkannya sesuai dengan panduan yang ada. Alasan dari deviasi tersebut bervariasi, beberapa bersifat pragmatis seperti "cara ini lebih efisien untuk tim" tetapi beberapa terpengaruh oleh kultur hierarkis perusahaan yang sudah tertanam sebelumnya [10].

Beberapa penelitian menunjukkan hasil yang berbeda terkait letak deviasi *Scrum* yang biasanya terjadi. Tetapi menurut [10] unsur yang paling sering terdeviasi adalah fungsi dan definisi peran, estimasi beban kerja, dan *quality assurance* sedangkan yang jarang terdampak adalah panjang *sprint*, rangkaian kegiatan, ukuran tim dan kebutuhan *engineering.*

C. *ScrumBut*

Pada tahun 2005, Bas Vodde memperkenalkan tes sederhana yang mampu menilai tingkat adopsi *Agile* dan *Scrum* di Nokia-Siemens Finland. Tes ini dikenal sebagai Nokia Scrum Test dan sering disebut juga sebagai "Scrum, But… Test" [8].

ScrumBut adalah penyebab berbagai tim dan organisasi tidak bisa mengambil keuntungan penuh dari *Scrum* dan tidak menyadari manfaat dari menggunakan *Scrum* dalam proses pengembangan perangkat lunak mereka. *ScrumBut* memiliki pola dan sintaksis yang membuatnya bisa dengan mudah teridentifikasi: (*ScrumBut*)(Alasan)(Jalan tengah). Contoh: (Kami menggunakan *Scrum*, tetapi)(Tenaga kerja hanya sedikit, ada *critical bug* yang harus segera diselesaikan)(Jadi kami tidak selalu bisa memenuhi *time box* yang ada) [5].

Hal ini disebabkan berbagai hal seperti, tingkat pemahaman dan keinginan mengimplementasikan *Scrum* yang minim ataupun peraturan, kultur dan birokrasi yang sudah ada yang tidak sejalan dengan nilai dan prinsip-prinsip *Scrum* yang kemudian diselesaikan dengan mencari "*workaorund"* di luar dari nilai dan prinsip yang ada.

D. *Anti-patterns*

Anti-patterns adalah sebuah pola yang biasanya berisi sebuah solusi dari suatu permasalahan, yang mana solusi tersebut hanya bersifat sementara, tidak benar-benar menyentuh inti permasalahannya yang kemudian justru berakibat negatif. *Anti-patterns* bisa merupakan hasil dari kurangnya pengetahuan anggota tim, sehingga tidak mampu menyelesaikan suatu permasalahan spesifik ataupun akibat dari penerapan suatu metode yang baik pada konteks yang salah. Ketika terdokumentasi dengan baik, suatu *anti-patterns* dapat mendeskripsikan suatu formula yang membantu kita untuk mengidentifikasi suatu deviasi, gejala yang mengindikasikan terjadinya suatu deviasi, konsekuensi dari deviasi itu sendiri dan solusi terhadap deviasi yang terjadi [7].

III. METODE

Seluruh data mengenai fenomena deviasi implementasi *Scrum* di Ralali dilakukan dengan menggunakan dua metode yaitu observasi dan wawancara yang kemudian akan dikomparasikan dengan teori-teori yang didapat dari studi literatur. Observasi dilakukan bersamaan dengan pelaksanaan kegiatan magang di Ralali. Keterlibatan langsung di dalam proses kerja tim *development* mampu memberikan perspektif dan hasil yang akurat. Yang kemudian diperkuat dengan melakukan wawancara kepada beberapa individu di dalam perusahaan, mencakup posisi project manager serta CTS dan technical program manager yang sudah memiliki sertifikasi scrum master.

Kumpulan deviasi yang didapat dari hasil observasi dan wawancara akan dipresentasikan menggunakan format *anti-patterns* seperti yang terlihat pada Table 2. Format ini berasal dari [7] yang kemudian dimodifikasi oleh [9]. Isi atau konteks dari *anti-patterns* ini sendiri disarikan dari hasil observasi dan wawancara yang telah dilakukan dengan penambahan dari literatur *Scrum* yang ada.

TABLE I. FORMAT ANTI-PATTERNS

Bidang Anti-pattern	Konten
Nama	Nama yang diberikan untuk *anti-pattern*

Bidang Anti-pattern	Konten
Rekomendasi *Scrum*	Rekomendasi dari literatur *Scrum* yang berhubungan
Konteks	Kondisi spesifik yang di mana *anti-pattern* sering muncul
Solusi	Solusi dari *anti-pattern* (yang biasanya berbahaya) atau deviasi *Scrum*nya itu sendiri
Konsekuensi	Konsekuensi dari menggunakan *anti-pattern*
Pengecualian	Kemungkinan keadaan yang membuat penggunaan *anti-pattern* bisa dibenarkan
Rekomendasi Perusahaan	Rekomendasi yang diambil dari hasil observasi dan wawancara yang dilakukan di dalam lingkungan perusahaan.

Nama dari *anti-patterns* harus berbentuk kata benda yang unik dan berkonotasi jelek [10]. Rekomendasi *Scrum* diambil dari literatur *Scrum* yang berhubungan. Berbeda dengan rekomendasi dari perusahaan yang berbentuk *best practice* yang didasarkan fakta empiris pada perusahaan tersebut. Konteks setiap *anti-patterns* diambil dari hasil kesimpulan observasi dan wawancara yang dilakukan, konteks ini menjelaskan kondisi spesifik di mana biasanya suatu *anti-patterns* bisa terjadi. Solusi merupakan solusi *anti-patterns* atau deviasi *Scrum* yang diambil sesuai dengan konteks yang ada. Konsekuensi dari penerapan *anti-patterns* sebagian diambil dari hasil observasi dan wawancara yang ada dan sebagian yang lain diambil dari literatur terkait [9].

Pengecualian adalah kondisi di mana suatu *anti-patterns* bisa dibenarkan penggunaannya menurut penulis [9]. Tetapi beberapa *anti-patterns* tidak memiliki pengecualian terlepas dari kondisi apa pun. Hal ini mengindikasikan bahwa menurut penulis tidak ada kondisi atau konteks yang membenarkan penggunaan *anti-patterns* tersebut dan hasil implementasi *anti-patterns* ini sepenuhnya bernilai negatif.

Rekomendasi berasal dari hasil observasi terkait deviasi yang dilakukan oleh Ralali dan rekomendasi solusinya jika ada pihak lain mengalami konteks *anti-patterns* yang sejenis.

IV. Pembahasan dan Hasil

A. Format Anti-patterns di Ralali

Berdasarkan hasil observasi dan wawancara, ada setidaknya tiga deviasi yang dilakukan oleh Ralali dari panduan *Scrum* yang ada. Tiga deviasi tersebut digambarkan dalam tiga tabel yaitu Table II Gangguan Eksternal, Table III *Sprint Pre-Review*, dan Table IV Variasi Waktu Sprint.

TABLE II. Gangguan Eksternal

Bidang Anti-pattern	Konten
Nama	Gangguan eksternal
Rekomendasi *Scrum*	Tidak ada pos kerja yang dibebankan kepada tim *Scrum* selain item yang sudah terdaftar pada *product backlog*.
Konteks	*Bug* atau sistem eror sering terjadi dan mempunyai *velocity* atau nilai prioritas sebesar 0. Mengindikasikan tidak ada prioritas yang lebih tinggi dari *existing bug.* Terlepas dari tugas apa pun yang tengah dikerjakan oleh tim *Scrum*, maka harus ditinggalkan jika mendapatkan sebuah *bug.*
Solusi	Menyelesaikan *bug* yang ada secepatnya dan memperpanjang estimasi waktu yang sudah ditentukan sebelumnya.
Konsekuensi	Sering berakhir dengan *carry-over*. Yaitu kondisi di mana item yang seharusnya diselesaikan pada *sprint* A, akhirnya dibawa ke *sprint* B karena belum mencapai *definition of done*. Estimasi waktu yang tidak sesuai prediksi awal.
Pengecualian	Tidak ada pengecualian.
Rekomendasi Perusahaan	Perusahaan atau tim *Scrum* memilih satu anggota tim yang hanya mengambil tiket bernilai rendah sehingga implikasi yang disebabkan penundaan pengerjaan tugas karena harus menyelesaikan suatu *bug* dapat lebih diminimalisasi. Atau bentuk satu tim khusus yang bertanggung jawab terhadap laporan *bug* yang terjadi.

TABLE III. *Sprint Pre-Review*

Bidang Anti-pattern	Konten
Nama	*Sprint Pre-Review*
Rekomendasi *Scrum*	Memaksimalkan *daily scrum* sebagai review internal sebelum melakukan *sprint review b*ersama *stake holder*.
Konteks	*Sprint review* gagal dikarenakan adanya kesalahan teknis, *requirement* yang belum tercapai, *bug* dan berbagai masalah lain.
Solusi	Tim *Scrum* menyelenggarakan *sprint pre-review* dua hari sebelum jadwal *sprint review*. Jika ada kegagalan sistem atau *requirement* yang belum terpenuhi maka akan segera diselesaikan sehari sebelum *sprint review.*
Konsekuensi	*Sprint review* akan berjalan dengan lebih matang dikarenakan sudah ada *sprint pre-*review sebelumnya. Tetapi waktu bagi tim *development* untuk menyelesaikan item yang ada pada *product backlog* berkurang satu hari.
Pengecualian	Deviasi ini bisa menjadi deviasi yang baik untuk diimplementasikan jika sebelumnya seluruh anggota tim *Scrum* sudah diberi pemahaman dan menyetujui konsekuensi yang ada dari deviasi ini.
Rekomendasi Perusahaan	Perusahaan sejak awal benar-benar memaksimalkan penggunaan *daily scrum* hingga titik maksimalnya. Dengan inspeksi yang dilakukan secara berkala dan sikap transparan tentang kemajuan tiket masing-masing maka tidak akan lagi dibutuhkan adanya *sprint pre-review* lagi.

TABLE IV. Variasi Waktu Sprint

Bidang Anti-pattern	Konten
Nama	Variasi waktu *sprint*
Rekomendasi *Scrum*	2 - 4 minggu
Konteks	Kurangnya disiplin dam pengalaman bagi tim *Scrum* yang baru terbentuk. Gangguan dari *stake holder* maupun kondisi sosial.
Solusi	Mengurai lebih kecil lagi item *product backlog* yang ada sehingga *sprint* bisa dilakukan dalam waktu kurang dari dua minggu.
Konsekuensi	Pengulangan beberapa *Scrum events* seperti *sprint planning*, *sprint review*, dan *sprint retrospective*. Sedangkan dengan *sprint* yang berlangsung selama dua

Bidang Anti-pattern	Konten
	minggu, *events* tersebut cukup terlaksana satu kali saja.
Pengecualian	Tidak ada pengecualian.
Rekomendasi Perusahaan	Melakukan eksperimen dengan berbagai variasi waktu *sprint* yang berbeda untuk melihat waktu *sprint* mana yang paling sesuai dalam berbagai situasi yang mungkin terjadi.

B. *Dampak Anti-patterns*

Ketiga *anti-patterns* di atas memberikan dampak signifikan pada panjang waktu sprint dan estimasi waktu pengerjaan tugas.

Disrupsi tiket *bug* yang mengambil porsi waktu pengerjaan tugas dan *sprint pre-review* yang juga mengambil satu hari kerja efektif membuat estimasi waktu pengerjaan menjadi tidak tentu membuat panjang waktu *sprint* menjadi bervariasi dan tidak menentu. Keseluruhan *anti-patterns* yang didapatkan saling terkait walau dasar permasalahan yang ada mungkin berbeda.

Anti-patterns tersebut ada yang tidak memiliki pengecualian yang berarti hingga titik ini belum ada pembenaran yang bisa diberikan bagi penerapan deviasi tersebut dalam lingkup kerja keseharian, contohnya gangguan eksternal. Tetapi ada juga *sprint pre-review* yang menyimpang dari rekomendasi *Scrum* yang ada, tetapi jika keseluruhan tim *Scrum* tidak keberatan, maka bisa diadopsi untuk proses *sprint* selanjutnya.

Penerapan suatu *anti-patterns* hanya bisa memberikan dampak positif ketika ada konteks yang mengharuskan deviasi dilakukan dengan analisis matang dan kesepakatan seluruh pihak yang terlibat sebelumnya. Pada kasus Ralali, dari tiga *anti-patterns* yang teridentifikasi hanya ada satu *anti-patterns Pre-Review* yang memiliki pengecualian dan dapat dibenarkan penggunaannya. Sedangkan *anti-patterns* Gangguan Eksternal dan Variasi Waktu *Sprint* masih bernilai negatif dan belum bisa dibenarkan penggunaannya.

V. BATASAN

Scrum tidak hanya cukup dinilai dari perspektif tim *development*. Melainkan harus ikut melibatkan keseluruhan organisasi, mengingat integrasi praktik *Scrum* tidak hanya terbatas di dalam lingkup *development* tetapi juga mencakup berbagai aspek aktivitas lain di dalam suatu perusahaan.

Di samping itu, makalah ini membutuhkan responden yang jauh lebih luas untuk dapat menghasilkan kesimpulan yang komprehensif dan berimbang. Baik dari segi umur perusahaan, lokasi geografis, latar belakang, sektor industri yang digeluti dan kultur yang ada pada perusahaan tersebut.

VI. KESIMPULAN

Terlepas nilai konsekuensi yang diberikan, *anti-patterns* yang ada masih tetap diterapkan dikarenakan beberapa alasan dan kondisi internal mereka seperti jumlah tenaga kerja yang terbatas yang menyebabkan *anti-patterns* Gangguan Eksternal terus terjadi dan tenaga kerja yang belum sepenuhnya memahami konsep *Scrum* yang membuat beberapa *tasks* tidak terselesaikan sehingga waktu *sprint* mengalami perpanjangan.

Dari pemaparan di atas, dapat ditarik beberapa kesimpulan sebagai berikut:

- Penggunaan *anti-patterns* untuk merumuskan deviasi *Scrum* bisa membantu untuk memetakan konteks atau latar belakang terjadinya deviasi, kapan suatu deviasi bisa dibenarkan dan bernilai positif serta solusi dan rekomendasi dari setiap deviasi yang terjadi.
- Cepatnya perubahan pada bidang rekayasa perangkat lunak membuat beberapa metode dan praktik baru, diterapkan oleh pelaku industri bahkan sebelum bukti kuat akan keabsahan teori dari metode dan praktik tersebut ada [9].
- Poin di atas mengakibatkan banyaknya deviasi yang terjadi dikarenakan teori yang ada masih dalam tahap pengembangan dan belum mencakup berbagai permasalahan yang terjadi di lapangan.
- Deviasi yang terjadi pada Ralali tidak selamanya memiliki konsekuensi negatif. Penyelenggaraan *sprint pre-review* merupakan sebuah deviasi yang diambil setelah melewati pertimbangan matang dan dilakukan oleh individu atau tim yang mempunyai pengetahuan cukup dan disepakati oleh seluruh pihak terkait sehingga mampu memberikan konsekuensi yang positif.

REFERENSI

[1] Agile Alliance. "Agile 101." Agilealliance.org. https://www.agilealliance.org/agile101/ (diakses Jun. 7, 2020).

[2] Agile Manifesto. "History: The Agile Manifesto." Agilemanifesto.org. https://agilemanifesto.org/history.html (diakses Jun. 7, 2020).

[3] I. Sommervile, *Software Engineering*, 10th ed. England: Pearson, 2016.

[4] K. Schwaber and J. Sutherland, *Software in 30 days: How agile managers beat the odds, delight their customers, and leave competitors in the dust*. New York, NY, USA: Wiley, 2012.

[5] Scrum. The Home of Scrum. "What is ScrumBut?" Scrum.org. https://www.scrum.org/resources/what-scrumbut (diakses Jun. 8, 2020).

[6] Scrum Guide. "The Scrum Guide." Scrumguides.org. https://www.scrumguides.org/scrum-guide.html (diakses Jun. 8, 2020).

[7] W. H. Brown, R. C. Malveau, H. W. McCormick III, T. J. Mowbray. *Refactoring Software, Architectures, and Projects in Crisis*. New York, NY, USA: Wiley, 1998.

[8] J. Sutherland. "ScrumButt Test aka Nokia Test." Jeffsutherland.com. http://jeffsutherland.com/scrum/ScrumButtTest.pdf (diakses Jun. 9, 2020)

[9] V. Eloranta, K. Koskimies, T. Mikkonen. "Exploring ScrumBut —An empirical study of Scrum anti-patterns." In: *Information and Software Technology*, vol. 74, pp. 194–203, June 2016

[10] P. Diebold, J. Ostberg, S. Wagner, U. Zendler. "What Do Practitioners Vary in Using Scrum?" In: *Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP*, vol. 212, pp. 40–51, May 2015