PT TELEKOMUNIKASI INDONESIA Tbk.

STANDAR SISTEM APLIKASI
SECURE AGILE SOFTWARE DEVELOPMENT

Nomor Dokumen	**:**	**STD E-003-2017**
Versi	**:**	**1.0**
Tanggal	**:**	**… Oktober 2017**

Diterbitkan oleh:
PT TELEKOMUNIKASI INDONESIA Tbk.
Divisi Digital Service
Jl. Geger Kalong Hilir No.47 Bandung 40152
Telepon : + 62 22 4574784
Faksimili : + 62 22 2014669

PT TELEKOMUNIKASI INDONESIA Tbk.

STANDAR SISTEM APLIKASI
SECURE AGILE SOFTWARE DEVELOPMENT

Nomor Dokumen	**:**	**STD E-003-2017**
Versi	**:**	**1.0**

Ditetapkan di	**:**	**Bandung**
Pada tanggal	**:**	**… Oktober 2017**

EGM DIVISI DIGITAL SERVICE

ARIEF MUSTAÍN
NIK. ……..

DAFTAR ISI

DAFTAR TABEL

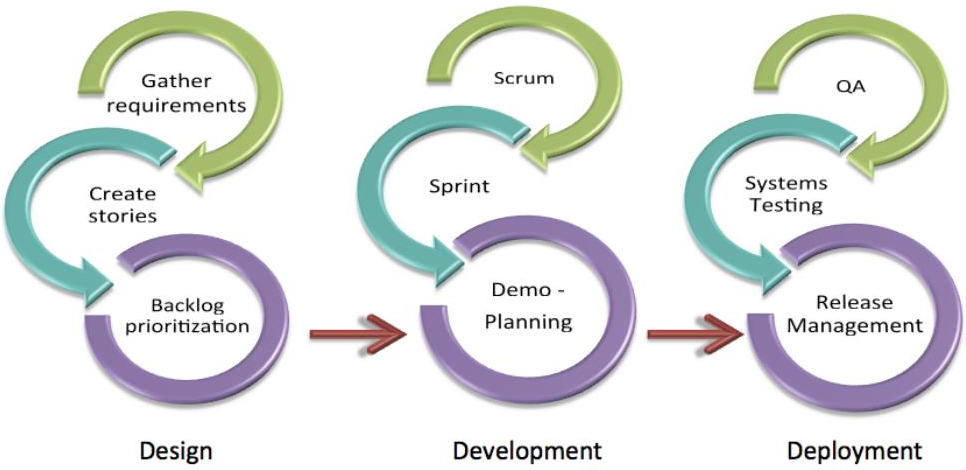
DAFTAR GAMBAR

1 UMUM

Standar *Secure Agile Software Development Lifecycle* ini dikembangkan untuk menjadi pedoman bagi pengembangan *software agile* yang wajib digunakan oleh *Amoeba*, *Startup* maupun *developer* TELKOM lainnya yang menggunakan pendekatan metodologi *Agile Software Development Lifecycle*

Secara umum, semua pengembangan aplikasi yang dilakukan dengan pendekatan Agile harus memenuhi hal-hal berikut:

1. Semua pengembangan aplikasi wajib memiliki fungsi *Secure Coding* dan bertanggung jawab terhadap persyaratan dan implementasi pada aplikasi yang dikembangkan.
2. Dalam setiap pengembangan aplikasi wajib memasukkan analisa dan persyaratan *Security* dalam setiap iterasi tahapan pengembangan yang dilakukan, dan diuji baik secara internal maupun dalam *acceptance test* untuk setiap rilis.
3. Penambahan aspek *IT Security* dalam praktis *Agile Software Development Lifecycle* layaknya Scrum, diwajibkan ada *item Product Backlog* yang berisi *Feature Stories* maupun *Abuse Feature Stories* dalam setiap iterasi *Sprint* pada aplikasi yang dikembangkan.



Gambar 1. Common Agile Software Development (Scrum)

1.1 Ruang Lingkup

Dokumen ini menjelaskan Standar *Secure Agile Software Development Lifecycle* yang dikembangkan secara khusus berdasarkan Scrum dan digunakan pada lingkungan TELKOM serta diimplementasikan oleh Tim *Amoeba*, *Startup*, maupun Tim lainnua yang menggunakan pendekatan serupa (Agile). Persyaratan yang diberikan dalam dokumen ini dapat mengalami perubahan sejalan dengan perkembangan kebutuhan TELKOM.

1.2 Deskripsi

Scrum merupakan sebuah kerangka kerja di mana orang-orang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, di mana pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif.

Scrum adalah kerangka kerja proses yang telah digunakan untuk mengelola pengembangan produk kompleks semenjak awal tahun 1990-an. Scrum bukanlah sebuah proses ataupun teknik untuk mengembangkan produk, lebih dari itu, scru merupakan sebuah kerangka kerja di mana di dalamnya dapat kita masukkan beragam proses dan teknik. Scrum akan mengekspos pergerakan efektifitas manajemen produk dan praktik pengembangan yang sedang dijalani, dengan begitu dapat diperoleh peningkatan.

Kerangka kerja Scrum terdiri dari Tim Scrum, serta peran-peran mereka di dalamnya; acara-acara; artefak-artefak; dan aturan-aturan. Setiap komponen di dalam kerangka kerja memiliki maksud tertentu dan peran penting demi keberhasilan penggunaan Scrum.

Aturan main Scrum menyatukan acara-acara, peran-peran dan artefak-artefak, menjaga harmonisasi dan interaksi antar setiap komponen. Dokumen ini menjelaskan aturan main Scrum secara umum.

2 PEDOMAN SECURE AGILE SOFTWARE DEVELOPMENT LIFECYCLE

2.1 Peran Utama dan Matrik Tanggung Jawab

Peran utama dalam *Secure Agile Software Development Lifecycle* (S-ASDL) di lingkungan TELKOM adalah sebagai berikut:

a. Unit *Product Owner Management* merupakan unit yang bertanggung jawab dalam mengkaji, membuat, mengembangkan berbagai produk digital yang menggunakan pendekatan *Agile*. Unit ini merupakan pengguna utama dokumen standar ini sebagai referensi intensif. (Catatan: Untuk saat ini, Unit *Product Owner Management* adalah semua Bidang di DDS yang bertanggung jawab mengembangkan aplikasi digital, para Amoeba, Startup, dan lainnya, serta dengan sendirinya menurut dokumen ini merupakan Tim Scrum).

b. Unit *Application Quality Assurance* merupakan unit yang bertanggung jawab dalam mengembangkan dan menetapkan standar kualitas serta melakukan serangkaian uji *Quality Assurance* berdasarkan standar kualitas tersebut untuk memastikan bahwa aplikasi yang dibuat memenuhi standar kualitas yang telah ditetapkan tersebut. (Catatan: untuk saat ini, *Unit Application Quality Assurance* adalah Bidang *Product & Infrastructure Assurance*).

c. Unit *Product Security Research & Policy* merupakan unit yang bertanggung jawab dalam mengkaji, membangun, mengembangkan dan menetapkan kebijakan keamanan pengembangan aplikasi dengan pendekatan *Agile* (*Secure Agile Software Development Lifecycle*). (Catatan: Untuk saat ini, *Unit Product Security Research & Policy* adalah Bidang *Infrastructure Research* & Standarization).

d. Unit *Digital Service Assurance* merupakan unit yang bertanggung jawab sebagai *touch point* bagi pengguna akhir aplikasi yang dikembangkan oleh Unit *Product Owner*. (Catatan: Untuk saat ini, Unit *Digital Product Service Assurance* adalah Bidang *Digital Service Assurance).*

Peran diatas bertanggung jawab dengan Matriks RACI sebagaimana tabel 1.

	POM	AQA	SRP	DSA
Penetapan kebijakan S-ASDLC	R	R,C	**A**,R	I
Pengujian Kualitas & Keamanan Aplikasi Major*	R	A,R	I,C	I
Pengujian Kualitas & Keamanan Aplikasi Minor*	A,R	R,C	I,C	I
Service Assurance	R	R	I,C	A,R
Integrasi Log Software Aplikasi ke Sistem Log Terpusat	R	I,R	I,C	A,R
Pengembangan Sistem Log Terpusat	I	R	A,R	R,C
Pengelolaan & Monitoring Operational Security Log	R	I,C	I,C	A,R

Tabel 1. Matriks RACI Secure Agile SDL

*catatan: pengujiah keamanan aplikasi minor maupun major disesuaikan dengan penomoran rilis versi aplikasi.

2.2 Pengaturan versi aplikasi

Untuk mempermudah dalam pengelolaan *software* aplikasi, terutama terkait rilis fitur, *troubleshooting*, dan lain-lain, maka pedoman yang digunakan sebagai berikut:

```
Major_Version.Minor_Version.Built_Number
```

Setiap siklus pengembangan versi (*Major*, *Minor*, *Built*) harus memasukkan kebutuhan keamanan di dalamnya. Sedangkan untuk penyederhanaan akuntabilitas pengujian, maka pengujian kualitas dan keamanan aplikasi diatur sesuai level versi. pengaturannya agar mengacu pada referensi matrik RACI Peran S-ADLC.

2.3 Ketentuan Dasar SCRUM

Scrum didasari oleh teori kontrol proses empiris, atau dengan kata lain, empirisme. Empirisme menekankan bahwa pengetahuan berasal dari pengalaman dan pembuatan keputusan didasari oleh pengetahuan yang telah dimiliki hingga saat ini. Scrum menggunakan pendekatan berkala (iterative) dan bertahap (incremental) untuk meningkatkan prediktabilitas dan mengendalikan resiko.

Ada tiga pilar dari setiap implementasi kontrol proses empiris yakni: transparansi, inspeksi dan adaptasi.

2.3.1 Transparansi

Aspek-aspek penting dari proses yang berjalan harus dapat ditinjau oleh pihak-pihak yang bertanggung-jawab terhadap hasilnya. Transparansi mengharuskan aspek-aspek tersebut didefinisikan dengan standar yang sama, sehingga semua peninjau memiliki pemahaman yang sama mengenai apa yang sedang ditinjau.

Sebagai contoh:

a. Istilah-istilah pada proses yang sedang digunakan harus dapat dimengerti oleh setiap pihak; dan,

b. Setiap pihak yang bekerja dan pihak yang menerima hasil pekerjaan harus memiliki pemahaman yang sama mengenai arti kata “Selesai”.

2.3.2 Inspeksi

Pengguna Scrum harus secara rutin meninjau artefak Scrum beserta perkembangannya agar perubahan dapat terdeteksi. Peninjauan hendaknya tidak terlalu sering sehingga dapat menyebabkan terhambatnya pekerjaan. Peninjauan paling bermanfaat jika dilakukan secara rutin, oleh peninjau yang kompeten, pada saat pekerjaan berjalan.

2.3.3 Adaptasi

Apabila peninjau mendapatkan satu atau lebih aspek dari proses mengalami deviasi di luar batasan yang dapat diterima, hingga hasil akhirnya menjadi tidak dapat diterima, maka proses atau materi yang diolah harus diatur ulang. Pengaturan ulang harus dibuat sesegera mungkin untuk meminimalisir deviasi yang lebih jauh.

Scrum menyediakan empat acara formal, yang dikumpulkan di dalam Sprint, untuk inspeksi dan adaptasi, sebagaimana dijelaskan di bagian Acara Scrum di dalam dokumen ini:

a. Sprint Planning
b. Daily Scrum
c. Sprint Review
d. Sprint Retrospective

2.4 Tim scrum

Tim Scrum merupakan bagian dari Product Owner Management dan terdiri atas Product Owne, Tim Pengembang dan Scrum Master. Tim Scrum mengatur diri mereka sendiri dan berfungsi antar-lintas. Tim yang mengatur dirinya sendiri menentukan cara terbaik untuk menyelesaikan pekerjaannya, daripada diatur oleh pihak lain yang berada di luar anggota tim. Tim yang berfungsi antar-lintas memiliki semua kompetensi yang dibutuhkan untuk menyelesaikan pekerjaan, tanpa mengandalkan pihak lain yang berada di luar anggota tim. Model tim di dalam Scrum dirancang sedemikian rupa untuk mengotimalisasi fleksibilitas, kreatifitas dan produktifitas. Setiap anggota Tim Scrum harus mendapatkan Security Education minimal sekali dalam kurun waktu setahun.

Tim Scrum menghantarkan produk secara berkala dan bertahap untuk memperbesar kesempatan mendapatkan masukan. Penghantaran secara bertahap dari sebuah produk yang “Selesai”, memastikan produk yang berpotensi dapat digunakan, selalu siap.

2.4.1 Product Owner

Product Owner bertanggung-jawab untuk memaksimalkan nilai produk dan hasil kerja Tim Pengembang. Cara pelaksanaannya sangat bervariasi antar organisasi, Tim Scrum dan individu.

Product Owner merupakan satu-satunya orang yang bertanggung-jawab untuk mengelola Product Backlog. Pengelolaan Product Backlog mencakup:

a. Mengekspresikan dengan jelas item Product Backlog;
b. Mengurutkan item di dalam Product Backlog untuk mencapai tujuan dan misi dengan cara terbaik;
c. Mengoptimalkan nilai dari hasil pekerjaan Tim Pengembang;
d. Memastikan Product Backlog transparan, jelas, dan dapat dilihat semua pihak, dan menunjukkan apa yang akan dikerjakan oleh Tim Scrum selanjutnya;
e. Memastikan Tim Pengembang dapat memahami item dalam Product Backlog hingga batasan yang diperlukan;
f. Identifikasi dan manajemen risiko keamanan pada item Product Backlog;
g. Menambahkan persyaratan privasi data pada item Product Backlog;
h. Mendefinisikan fitur-fitur keamanan pada item Product Backlog;

i. Mendefinisikan penyalahgunaan fitur yang dapat dilakukan oleh pengguna (abuse stories) pada item Product Backlog;

Product Owner dapat saja mengerjakan pekerjaan-pekerjaan di atas, atau menyerahkan pengerjaannya kepada Tim Pengembang, namun satu-satunya pihak yang bertanggung jawab tetaplah Product Owner.

Product Owner adalah satu orang dan bukan berupa sebuah komite. Product Owner dapat mengejawantahkan aspirasi dari komite ke dalam Product Backlog, namun mereka yang ingin merubah prioritas item Product Backlog, harus melakukannya melalui Product Owner.

Agar Product Owner berhasil menjalankan tugasnya, seluruh organisasi harus menghormati setiap keputusan yang ia buat. Keputusan dari Product Owner ini dapat dilihat dari isi dan urutan Product Backlog. Tidak ada seseorang pun yang dapat memerintah Tim Pengembang untuk mengerjakan kebutuhan lain selain Product Owner. Dan Tim Pengembang pun tidak diperbolehkan untuk melakukan apa yang diperintahkan oleh pihak lain selain Product Owner.

2.4.2 Tim Pengembang

Tim Pengembang terdiri dari para profesional yang bekerja untuk menghasilkan tambahan potongan produk (selanjutnya disebut Inkremen) “Selesai”, yang berpotensi untuk dirilis di setiap akhir Sprint. Hanya anggota Tim Pengembang yang mengembangkan Inkremen ini.

Tim Pengembang dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang ada di dalam tim akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan.

Tim Pengembang memiliki karakteristik sebagai berikut:

a. Mereka mengatur dirinya sendiri. Tidak ada satu orang pun (bahkan Scrum Master) yang memerintah Tim Pengembang bagaimana cara merubah Product Backlog menjadi Inkremen yang berpotensi untuk dirilis;

b. Tim Pengembang berfungsi antar-lintas, sebagai sebuah tim, memiliki semua keahlian yang dibutuhkan untuk menghasilkan produk;

c. Scrum tidak mengenal adanya jabatan tertentu untuk anggota Tim Pengembang selain Pengembang, apapun pekerjaan yang dikerjakan oleh masing-masing anggota tim; tidak ada pengecualian untuk aturan yang satu ini;

d. Tim Pengembang tidak mengenal adanya sub-tim yang dikhususkan untuk bidang tertentu seperti pengujian atau analisa bisnis; tidak ada pengecualian untuk aturan yang satu ini;

e. Anggota Tim Pengembang boleh memiliki spesialisasi keahlian dan fokus di satu area tertentu, namun akuntabilitas dari hasil dari pekerjaan secara keseluruhan adalah milik Tim Pengembang.

f. Tim Pengembang memodelkan ancaman (threats) yang mungkin terjadi pada produk.

g. Tim Pengembang wajib mengimplementasikan teknik kode yang aman (secure coding);

h. Tim Pengembang mengulas keamanan kode yang telah ditulis.

i. Tim Pengembang melakukan pengujian kode secara manual maupun otomatis.

j. Tim Pengembang mendokumentasikan item Product Backlog dengan Version Control System (VCS).

Jumlah anggota Tim Pengembang yang optimal adalah cukup kecil untuk dapat berkoordinasi dengan cepat, dan cukup besar untuk dapat menyelesaikan pekerjaan dalam satu Sprint. Jumlah anggota tim yang kurang dari tiga orang akan mengurangi interaksi dan akan menyebabkan produktifitas yang rendah. Tim Pengembang yang kecil kemungkinan akan mengalami kekurangan keahlian tertentu pada saat Sprint berjalan, yang pada akhirnya menyebabkan Tim Pengembang tidak dapat menghasilkan Inkremen yang berpotensi untuk dirilis. Tim Pengembang dengan jumlah anggota lebih dari sembilan orang membutuhan terlalu banyak koordinasi. Tim Pengembang dengan jumlah anggota tim yang banyak, akan menimbulkan terlalu banyak kompleksitas bagi proses yang berbasiskan empirisme. Product Owner dan Scrum Master tidak termasuk dalam hitungan, kecuali mereka juga turut ikut mengerjakan pekerjaan yang ada di Sprint Backlog.

2.4.3 Scrum Master

Scrum Master bertanggung jawab untuk memastikan Scrum telah dipahami dan dilaksanakan. Scrum Master melakukannya dengan memastikan Tim Scrum mengikuti teori, praktik, dan aturan main Scrum.

Scrum Master adalah seorang pemimpin yang melayani Tim Scrum. Scrum Master membantu pihak di luar Tim Scrum, untuk memahami apakah interaksi mereka dengan Tim Scrum bermanfaat atau tidak. Scrum Master membantu setiap pihak untuk merubah interaksi-interaksi yang tidak bermanfaat sehingga bisa memaksimalkan nilai yang dihasilkan oleh Tim Scrum.

2.4.3.1 Layanan Scrum Master kepada Product Owner

Scrum Master melayani Product Owner dengan berbagai cara yang mencakup:

a. Mencari teknik yang paling efektif untuk mengelola Product Backlog;

b. Membantu Tim Scrum untuk memahami pentingnya Product Backlog item yang jelas dan padat;

c. Memahami bagaimana perencanaan produk pada lingkungan yang didasarkan empirisme;

d. Memastikan Product Owner tahu bagaimana mengelola Product Backlog guna memaksimalkan nilai dari produk;

e. Memahami dan mempraktikkan agility; dan,

f. Memfasilitasi acara-acara dalam Scrum bila dipanggil dan dibutuhkan.

2.4.3.2 Layanan Scrum Master kepada Tim Pengembang

Scrum Master melayani Tim Pengembang lewat berbagai cara yang mencakup:

a. Membimbing Tim Pengembang untuk dapat mengatur dirinya sendiri dan berfungsi antar-lintas;

b. Membantu Tim Pengembang untuk membuat produk bernilai tinggi;

c. Menghilangkan hambatan-hambatan yang dialami oleh Tim Pengembang;

d. Memfasilitasi acara-acara dalam Scrum bila dipanggil dan dibutuhkan; dan,

e. Membimbing Tim Pengembang dalam suasana organisasi di mana Scrum belum sepenuhnya diterapkan dan dipahami.

2.4.3.3 Layanan Scrum Master kepada Organisasi

Scrum Master melayani organisasi tempat dia berada lewat berbagai cara yang mencakup:

a. Memimpin dan membimbing organisasi dalam penerapan Scrum;

b. Merencanakan implementasi Scrum di dalam organisasi;

c. Membantu setiap pegawai dan stakeholder dalam memahami dan menggunakan Scrum dan pengembangan produk dengan metoda empiris;

d. Membuat perubahan yang dapat meningkatkan produktifitas di dalam Tim Scrum; dan,

e. Bekerja bersama dengan Scrum Master lainnya guna meningkatkan efektifitas dari pengaplikasian Scrum di dalam organisasi.

2.5 Acara-acara Scrum

Acara-acara wajib dalam Scrum dihadiri untuk menciptakan sebuah kesinambungan dan mengurangi adanya acara-acara lain yang tidak tercantum di dalam Scrum. Setiap acara di dalam Scrum memiliki batasan waktu, yang artinya selalu memiliki durasi maksimum. Pada saat Sprint dimulai, durasinya tetap dan tidak dapat diperpendek maupun diperpanjang. Acara-acara lainnya dapat diakhiri saat tujuan dari acara tersebut telah tercapai; memastikan waktu digunakan secukupnya tanpa ada yang terbuang sia-sia di sepanjang proses.

Selain Sprint itu sendiri, yang memang merupakan kontainer dari acara-acara lain, setiap acara dalam Scrum adalah sebuah kesempatan formal untuk meninjau dan merubah sesuatu. Acara-acara ini dirancang secara khusus untuk menciptakan tranparansi dan peninjauan sampai ke tingkatan kritis. Tidak adanya pelaksanaan salah satu acara ini akan mengurangi transparansi dan menghilangkan kesempatan untuk meninjau dan membuat perubahan.

2.5.1 Sprint

Jantung dari Scrum adalah Sprint, sebuah batasan waktu selama satu bulan atau kurang, di mana sebuah Inkremen yang “Selesai”, berfungsi, berpotensi untuk dirilis dikembangkan. Sprint biasanya memiliki durasi yang konsisten sepanjang proses pengembangan produk. Sprint yang baru, langsung dimulai setelah Sprint yang sebelumnya berakhir.

Sprint memuat dan terdiri dari Sprint Planning, Daily Scrum, pengembangan, Sprint Review dan Sprint Retrospective.

Pada saat Sprint:

a. Tidak boleh ada perubahan yang dapat membahayakan tercapainya Sprint Goal;

b. Kualitas dari Sprint Goal tidak boleh menurun;

c. Scope dapat diklarifikasikan dan dinegosiasikan ulang diantara Product Owner dan Tim Pengembang seiring dengan bertambahnya pengetahuan.

d. Tim Pengembang menggunakan unit test untuk membuktikan fungsi kode yang dikirimkan sesuai yang dirancang. Unit test dijalankan setelah berhasil dideploy dan membantu menangkap bug yang disebabkan oleh perubahan terbaru. Unit Test sering mencakup serangan dan injeksi untuk memastikan bahwa aplikasi tahan terhadap masalah potensial yang diuraikan selama pemodelan ancaman.

e. Tim Pengembang melakukan uji regresi keamanan: Tes regresi memverifikasi bahwa perubahan kode benar-benar memperbaiki bug. Tetapi tidak seperti Unit Test, setiap uji regresi menargetkan adanya defek yang diketahui baik dalam kode yang sedang dikembangkan atau dalam modul pendukung.

f. Tim Pengembang memeriksa kode secara manual. Ulasan kode disebut juga peer review, dimana salah satu anggota Tim Pengembang memeriksa kode pengembang lain. Mereka memastikan kode sesuai dengan standar umum namun juga mencari kekurangan implementasi yang spesifik seperti variabel masukan yang tidak difilter, otentikasi pengguna yang tidak memadai, dan kesalahan yang tidak tertangani.

Setiap Sprint dapat dikatakan sebagai sebuah proyek dengan batasan waktu tidak lebih dari satu bulan. Sama halnya dengan proyek, Sprint digunakan untuk menyelesaikan sesuatu. Setiap Sprint memiliki definisi mengenai apa yang akan dikembangkan, sebuah disain dan perencanaan yang fleksibel yang akan membimbing pengembangan, pekerjaan yang akan dilakukan dan hasil dari produk.

Sprint dibatasi pada satu bulan kalender. Bila jangka waktu Sprint terlalu panjang, maka definisi mengenai apa yang akan dibangun dapat berubah, kompleksitas dapat meningkat, dan resiko dapat bertambah. Sprint meningkatkan prediktabilitas karena adanya peninjauan dan pengadaptasian terhadap perkembangan, setidaknya setiap satu bulan sekali. Sprint juga membatasi resiko biaya hingga satu bulan saja.

Membatalkan Sprint

Sprint dapat dibatalkan sebelum batasan waktu Sprint selesai. Hanya Product Owner yang dapat membatalkan Sprint, walaupun keputusan yang dia buat mungkin saja dipengaruhi oleh para stakeholder, Tim Pengembang, ataupun Scrum Master.

Sprint dibatalkan apabila Sprint Goal sudah tidak sesuai harapan mula-mula. Hal ini dapat terjadi apabila arahan perusahaan berubah, atau bila kondisi pasar atau teknologi berubah. Pada umumnya, Sprint harus dibatalkan apabila Sprint menjadi tidak masuk akal lagi apabila dilanjutkan. Namun karena batasan waktu Sprint yang begitu singkat, pembatalan biasanya jarang terjadi.

Ketika Sprint dibatalkan, item Product Backlog yang “Selesai” ditinjau kembali. Apabila hasil pekerjaan dari Product Backlog tersebut berpotensi untuk dirilis, biasanya Product Owner akan menerima hasil pekerjaan tersebut. Semua item Product Backlog yang tidak selesai, diestimasi dan dimasukkan kembali ke dalam Product Backlog.

Pekerjaan dalam Product Backlog yang sudah selesai tersebut akan mengalami depresiasi nilai dengan cepat dan harus diestimasi ulang sesering mungkin.

Pembatalan Sprint membuang banyak tenaga, karena semua orang harus menyusun kembali kelompoknya dalam Sprint Planning baru untuk memulai Sprint baru. Pembatalan Sprint sering menyebabkan trauma bagi Tim Scrum, dan sangat jarang terjadi.

2.5.2 Sprint Planning

Pekerjaan yang akan dilaksanakan di dalam Sprint direncanakan pada saat Sprint Planning. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota Tim Scrum. Sprint Planning dibatasi maksimum delapan jam untuk Sprint yang berdurasi satu bulan. Untuk Sprint yang lebih pendek, batasan waktunya biasanya lebih singkat. Scrum Master memastikan bahwa acara ini dilaksanakan dan setiap hadirin memahami tujuannya. Scrum Master mengedukasi Tim Scrum untuk melaksanakannya dalam batasan waktu yang telah ditentukan.

Sprint Planning harus dapat menjawab pertanyaan-pertanyaan berikut:

a. Apa goal dari Sprint?

b. Apa yang dapat dihantarkan di dalam Inkremen sebagai hasil dari Sprint yang sedang berjalan?

c. Apa yang perlu dilakukan untuk dapat menghantarkan Inkremen tersebut?

Topik Pertama: Apa yang dapat dilakukan di dalam Sprint ini?

Tim Pengembang bekerja untuk memperkirakan fungsionalitas yang akan dikembangkan pada saat Sprint. Product Owner menjabarkan obyektif yang harus dicapai di Sprint ini dan item Product Backlog mana, yang apabila bisa diselesaikan di Sprint ini, akan mencapai Sprint Goal.

Seluruh anggota Tim Scrum berkolaborasi untuk memahami pekerjaan di dalam Sprint. Masukan dari acara ini adalah Product Backlog, Inkremen yang terakhir, proyeksi kapasitas Tim Pengembang dalam satu Sprint, dan histori performa Tim Pengembang. Jumlah item yang dipilih dari Product Backlog untuk Sprint sepenuhnya diserahkan kepada Tim Pengembang. Hanya Tim Pengembang yang dapat menilai seberapa banyak item yang dapat diselesaikan di Sprint ini.

Setelah Tim Pengembang memperkirakan item Product Backlog yang akan selesai dan dihantarkan dalam Sprint ini, Tim Scrum mulai membuat Sprint Goal. Sprint Goal dapat menciptakan sebuah keselarasan di dalam pekerjaan Tim Pengembang, yang mungkin tidak akan ada bila masing-masing anggota tim memiliki inisiatif sendiri-sendiri tanpa adanya tujuan yang sama.

Topik Kedua: Bagaimana pekerjaan yang telah dipilih dapat diselesaikan?

Setelah Sprint Goal dibuat dan item Product Backlog dipilih, Tim Pengembang menentukan bagaimana mengembangkan fungsionalitas ini menjadi sebuah Inkremen yang “Selesai” pada saat Sprint. Item Product Backlog yang telah dipilih untuk Sprint ini beserta rencana sampai bisa selesai dan dihantarkan disebut sebagai Sprint Backlog.

Tim Pengembang biasanya memulai dengan merancang sistem dan pekerjaan yang perlu dilakukan untuk menjadikan Product Backlog menjadi Inkremen yang berfungsi penuh. Pekerjaan yang dirancang mungkin akan memiliki ukuran atau estimasi yang

berbeda-beda. Walaupun demikian, jumlah pekerjaan yang direncanakan pada saat Sprint Planning cukup banyak untuk dikerjakan selama satu Sprint. Pekerjaan yang direncanakan untuk hari-hari pertama dari Sprint dibagi-bagi jadi bagian-bagian kecil pada akhir acara ini, biasanya dalam satuan satu hari atau kurang. Tim Pengembang mengatur dirinya sendiri untuk mengambil pekerjaan di dalam Sprint Backlog, baik pada saat Sprint Planning maupun sepanjang Sprint.

Pada saat Tim Pengembang membuat perencanaan, rencana tersebut selalu mengacu pada Sprint Goal. Pada saat Sprint berjalan, pekerjaan yang harus dilakukan terkadang berbeda dengan apa yang telah direncanakan oleh Tim Pengembang pada saat Sprint Planning. Tim Pengembang akan berkolaborasi dengan Product Owner untuk menentukan cara terbaik untuk merevisi perencanaan dengan tetap mencapai Sprint Goal. Sprint Goal menyediakan fleksibilitas mengenai bagaimana fungsionalitas dapat diimplementasikan sebelum Sprint berakhir.

Product Owner dapat membantu mengklarifikasi item Product Backlog yang dipilih dan membuat pengecualian. Apabila Tim Pengembang mengatakan mereka memiliki terlalu banyak atau terlalu sedikit pekerjaan, mereka dapat menegosiasikan ulang item Product Backlog yang telah dipilih dengan Product Owner. Tim Pengembang juga dapat mengundang pihak lain untuk menghadiri acara ini guna memberikan masukan yang berhubungan dengan hal teknis ataupun domain permasalahan.

Di akhir Sprint Planing, Tim Pengembang sudah harus dapat menjelaskan kepada Product Owner ataupun Scrum Master, bagaimana mereka berencana untuk bekerja sebagai tim yang mengatur dirinya sendiri untuk menyelesaikan Sprint Goal, dan membuat Inkremen yang telah diantisipasi.

Untuk bagian keamanan yang dapat dilakukan antara lain:

a. Memperbaharui threat model.
b. Merencanakan studi kasus penyalahgunaan yang dilakukan oleh pengguna (abuse user stories).
c. Merencanakan fitur-fitur keamanan pada produk.
d. Mendefinisikan kriteria keamanan yang dapat diterima.
e. Mendefinisikan dan merencanakan strategi pengujian keamanan.

Sprint Goal

Sprint Goal adalah sekumpulan tujuan yang akan dicapai dalam satu Sprint sepanjang pengimplementasian Product Backlog. Sprint Goal memberikan arahan bagi Tim Pengembang mengapa mereka mengembangkan Inkremen dalam Sprint tersebut. Sprint Goal dibuat pada saat Sprint Planning. Sprint Goal memberikan Tim Pengembang fleksibilitas terkait bagaimana implementasi fungsionalitas di tengah Sprint. Item-item Product Backlog yang terpilih menghantarkan pada satu fungsionalitas yang selaras. Di mana bisa berupa Sprint Goal itu sendiri. Sprint goal bisa juga berupa fungsionalitas yang selaras apapun, yang pada akhirnya membuat Tim Pengembang berkerja bersama alih-alih dengan inisiatif sendiri-sendiri.

Tim Pengembang berkerja dengan dipandu oleh Sprint Goal. Untuk memenuhi Sprint Goal, mereka mengimplementasikan fungsionalitas & teknologi. Jika hasil kerja mereka ternyata berbeda dengan yang mereka duga sebelumnya, mereka berkolaborasi

dengan Product Owner untuk menegosiasikan ruang lingkup dari Sprint Backlog pada suatu Sprint.

2.5.3 Daily Scrum

Daily Scrum adalah kegiatan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam ke depan. Hal ini dilakukan dengan meninjau pekerjaan semenjak acara Daily Scrum terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan Daily Scrum berikutnya.

Daily Scrum dilaksanakan pada waktu dan tempat yang sama setiap hari untuk mengurangi kompleksitas. Pada saat pertemuan, Tim Pengembang menjelaskan:

a. Apa yang sudah saya lakukan kemarin yang telah membantu Tim Pengembang mencapai Sprint Goal?
b. Apa yang akan saya lakukan hari ini untuk membantu Tim Pengembang mencapai Sprint Goal?
c. Apakah ada hambatan yang dapat menghalangi saya atau Tim Pengembang untuk mencapai Sprint Goal?

Tim Pengembang menggunakan Daily Scrum untuk meninjau perkembangan menuju Sprint Goal dan meninjau tren perkembangan menuju selesainya pekerjaan yang ada di dalam Sprint Backlog. Daily Scrum mengoptimalisasi kemungkinan Tim Pengembang akan mencapai Sprint Goal. Setiap hari, Tim Pengembang harus memahami bagaimana caranya agar mereka dapat bekerja bersama sebagai tim yang mengatur dirinya sendiri, untuk menyelesaikan Sprint Goal, dan membuat Inkremen yang sudah diharapkan di akhir Sprint. Tim Pengembang atau beberapa anggota tim seringkali langsung bertemu setelah Daily Scrum untuk diskusi yang detail, atau untuk pengadaptasian, atau perubahan perencanaan, sisa pekerjaan dalam Sprint.

Scrum Master memastikan pertemuan ini berlangsung, namun yang bertanggung-jawab untuk melangsungkannya adalah Tim Pengembang. Scrum Master mengajarkan Tim Pengembang untuk melangsungkan Daily Scrum tidak lebih dari 15 menit.

Scrum Master memastikan bahwa hanya anggota Tim Pengembang yang berpartisipasi pada saat Daily Scrum.

Daily Scrum meningkatkan komunikasi, menghilangkan pertemuan-pertemuan lain, mengidentifikasi hambatan untuk dihilangkan, mendukung pembuatan keputusan secara cepat dan meningkatkan tingkat pengetahuan tim. Pertemuan ini adalah kunci dari proses peninjauan dan pengadaptasian.

Untuk bagian keamanan yang dapat dilakukan antara lain:

a. Mendiskusikan risiko keamanan produk.
b. Merencanakan ulang keamanan produk.

2.5.4 Sprint Review

Sprint Review diadakan di akhir Sprint untuk meninjau Inkremen dan merubah Product Backlog bila diperlukan. Pada saat Sprint Review, Tim Scrum dan stakeholder

berkolaborasi untuk membahas apa yang telah dikerjakan dalam Sprint yang baru usai. Berdasarkan hasil tersebut tersebut dan semua perubahan Product Backlog pada saat Sprint, para hadirin berkolaborasi menentukan apa yang dapat dikerjakan di Sprint berikutnya, untuk mengoptimalisasi nilai produk. Pertemuan ini bersifat informal, bukan merupakan status meeting, dan presentasi dari Inkremen diharapkan dapat mengumpulkan masukan dan menumbuhkan semangat kolaborasi.

Ini adalah acara dengan batasan waktu maksimum selama empat jam untuk Sprint yang berdurasi satu bulan. Untuk Sprint yang lebih pendek, batasan waktunya biasanya lebih singkat. Scrum Master memastikan bahwa acara ini dilaksanakan, dan setiap hadirin memahami tujuannya. Scrum Master mengedukasi Tim Scrum untuk melaksanakannya dalam batasan waktu yang telah ditentukan.

Sprint Review mencakup elemen-elemen berikut:

a. Hadirin termasuk Tim Scrum dan stakeholder kunci diundang oleh Product Owner;
b. Product Owner menjelaskan item Product Backlog apa yang sudah “Selesai” dan apa yang belum “Selesai”;
c. Tim Pengembang menjelaskan apa yang berjalan dengan baik sepanjang Sprint, masalah apa yang mereka hadapi, dan bagaimana mereka menyelesaikan masalah tersebut;
d. Tim Pengembang mendemonstrasikan pekerjaan yang sudah mereka “selesai”-kan dan menjawab pertanyaan-pertanyaan mengenai potongan tambahan produk;
e. Product Owner menjelaskan keadaan terakhir Product Backlog. Ia dapat memproyeksikan tanggal perkiraan selesai produk (bila dibutuhkan);
f. Seluruh hadirin berkolaborasi membahas pekerjaan selanjutnya, dengan begitu Sprint Review menyediakan masukan yang berarti bagi Sprint Planning berikutnya;
g. Ulasan mengenai keadaan pasar--atau kemungkinan potensi penggunaan produk--yang telah berubah dan hal yang paling berharga apa yang harus dikerjakan berikutnya; dan,
h. Review timeline, budget, potensi kapabilitas dan marketplace untuk antisipasi rilis produk.

Hasil dari Sprint Review adalah revisi dari Product Backlog yang mendefinisikan kemungkinan item Product Backlog untuk Sprint berikutnya. Product Backlog dapat dirubah secara keseluruhan sebagai tanggapan atas peluang-peluang baru.

2.5.5 Sprint Retrospective

Sprint Retrospective adalah sebuah kesempatan bagi Tim Scrum untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di Sprint berikutnya.

Sprint Retrospective dilangsungkan setelah Sprint Review selesai dan sebelum Sprint Planning berikutnya. Ini adalah acara dengan batasan waktu maksimum selama tiga jam untuk Sprint yang berdurasi satu bulan. Untuk Sprint yang lebih pendek, batasan waktunya biasanya lebih singkat. Scrum Master memastikan bahwa acara ini dilaksanakan dan setiap hadirin memahami tujuannya. Scrum Master mengedukasi Tim Scrum untuk melaksanakannya dalam batasan waktu yang telah ditentukan.

Scrum Master berpartisipasi sebagai rekan yang bertanggung-jawab terhadap proses Scrum.

Tujuan dari Sprint Retrospective adalah:

a. Meninjau bagaimana Sprint yang telah selesai berlangsung, termasuk hal-hal yang berkaitan dengan orang-orangnya, hubungan antara orang-orang, proses, dan perangkat kerja;
b. Mengidentifikasi dan mengurutkan hal-hal utama yang berjalan baik, dan hal-hal yang berpotensi untuk ditingkatkan; dan,
c. Membuat rencana implementasi, dengan tujuan peningkatan cara-cara kerja Tim Scrum.

Scrum Master mengedukasi Tim Scrum untuk membuat peningkatan akan kerangka kerja proses Scrum, juga proses dan praktik pengembangannya, sehingga lebih efektif dan menyenangkan di Sprint berikutnya. Pada saat Sprint Retrospective, Tim Scrum merencanakan cara untuk meningkatkan kualitas dari produk, dengan merubah definisi dari “Selesai” sebagaimana dibutuhkan.

Di akhir Sprint Retrospective, Tim Scrum harus dapat mengidentifikasi peningkatan-peningkatan yang akan diimplementasikan di Sprint berikutnya. Mengimplementasikan peningkatan ini di Sprint berikutnya, merupakan salah satu bentuk adaptasi dari hasil peninjauan Tim Scrum itu sendiri. Walaupun peningkatan-peningkatan dapat diimplementasikan kapanpun juga, Sprint Retrospective memberikan kesempatan formal untuk fokus pada peninjauan dan adaptasi.

Untuk bagian keamanan yang dapat dilakukan pada Sprint Review dan Retrospective antara lain:

a. Verifikasi pemodelan ancaman (threat) secara kesuluruhan.
b. Mengulas penyalahgunaan yang dilakukan oleh pengguna (abuse user stories) dan juga mengulas kriteria keamanan yang dapat diterima.
c. Memberikan himbauan keamanan untuk pelanggan produk.
d. Menginspeksi dan mengadaptasi aktivitas keamanan produk bisa melalui (log) dan lain-lain.

2.6 Artefak Scrum

Artefak Scrum merepresentasikan pekerjaan atau nilai, bertujuan untuk menyediakan transparansi, dan kesempatan-kesempatan untuk peninjauan dan adaptasi. Artefak yang didefinisikan oleh Scrum secara khusus dirancang untuk meningkatkan transparansi dari informasi kunci, dengan begitu semua pihak dapat memiliki pemahaman yang sama terhadap artefak.

2.6.1 Product Backlog

Product Backlog adalah daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. Product Owner bertanggung-jawab terhadap Product Backlog, termasuk isinya, ketersediaannya, dan urutannya.

Product Backlog tidak pernah selesai. Pada awal pembuatannya hanya terjabar daftar kebutuhan yang paling diketahui dan dipahami pada saat itu. Product Backlog berkembang seiring dengan berkembangnya produk dan lingkungan di mana produk tersebut digunakan. Product Backlog bersifat dinamis; senantiasa berubah agar produk dapat menjadi layak, kompetitif di pasar, dan bermanfaat bagi penggunanya. Selama produk masih eksis maka Product Backlog juga eksis.

Product Backlog menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan dan perbaikan terhadap produk di rilis mendatang. Item Product Backlog memiliki atribut deskripsi, urutan, estimasi dan nilai bisnis.

Seiring dengan digunakannya produk dan semakin bertambahnya nilai dari produk, dan bertambahnya masukan dari pasar, Product Backlog semakin berkembang menjadi lebih besar. Daftar kebutuhan tidak pernah berhenti berubah, sehingga Product Backlog dapat dikatakan sebagai artefak yang hidup. Perubahan dalam kebutuhan bisnis, keadaan pasar, ataupun teknologi dapat menyebabkan perubahan pada Product Backlog.

Tidak jarang ditemukan lebih dari satu Tim Scrum mengerjakan satu produk yang sama. Satu Product Backlog digunakan untuk menggambarkan pekerjaan selanjutnya terhadap sebuah produk. Bisa ditambahkan sebuah atribut, untuk mengelompokkan item Product Backlog.

Product Backlog refinement adalah kegiatan menambahkan detail, mengestimasi dan mengurutkan item di dalam Product Backlog. Kegiatan ini berkesinambungan, di mana Product Owner dan Tim Pengembang berkolaborasi untuk merinci item Product Backlog. Pada saat Product Backlog refinement, item ditinjau-ulang dan direvisi. Tim Scrum sendiri yang menentukan bagaimana dan kapan proses refinement diadakan. Refinement biasanya memakan tidak lebih dari 10% kapasitas Tim Pengembang. Walaupun demikian, item Product Backlog dapat diperbarui kapanpun juga oleh Product Owner--atau siapapun atas arahan Product Owner--kapanpun ia mau.

Item Product Backlog pada urutan yang lebih atas biasanya lebih jelas dan lebih detail dibandingkan item di bawahnya. Estimasi dengan presisi tinggi diberikan berdasarkan tingkat kejelasan dan detail yang tinggi; semakin bawah urutan dari item Product Backlog, maka semakin rendah pula tingkat kedetailannya. Item Product Backlog yang akan dikerjakan oleh Tim Pengembang untuk Sprint yang mendatang di-refine supaya setiap item yang dikerjakan dapat di-“Selesai”-kan dalam satu Sprint. Item Product Backlog yang dianggap dapat di-“Selesai”-kan oleh Tim Pengembang dalam satu Sprint dikatakan “Siap” untuk diseleksi pada saat Sprint Planning. Item Product Backlog biasanya akan memiliki tingkat transparansi yang tinggi karena adanya aktifitas refinement ini.

Tim Pengembang bertanggung-jawab terhadap seluruh estimasi. Product Owner dapat mempengaruhi Tim Pengembang dengan cara membantu mereka memahami Product Backlog dan membuat pengecualian terhadap Product Backlog, namun orang-orang yang akan mengerjakan item Product Backlog-lah yang akan membuat estimasi final.

Memantau perkembangan menuju Sprint Goal

Di titik manapun, jumlah pekerjaan yang tersisa hingga akhir tujuan pengembangan dapat dijumlahkan. Product Owner memantau total sisa pekerjaan ini setidaknya di setiap Sprint Review. Product Owner membandingkan kondisi saat ini dengan jumlah sisa pekerjaan di Sprint Review sebelumnya guna meninjau perkembangan menuju

tujuan akhir dengan waktu yang diharapkan. Informasi ini transparan untuk setiap stakeholder.

Berbagai macam praktik proyeksi terhadap trending telah digunakan untuk memperkirakan kemajuan, seperti burn-down atau burn-up. Hal ini telah terbukti berguna. Namun hal ini tidak menggantikan pentingnya peran empirisme. Di dalam lingkungan yang kompleks, apa yang akan terjadi di masa mendatang tidak dapat diketahui sebelumnya. Hanya apa yang telah terjadi yang dapat digunakan untuk membuat keputusan di masa mendatang.

Hal-hal yang harus diperhatikan untuk Product Backlog pada sisi Keamanan antara lain:

a. Pemodelan ancaman dapat dilakukan dengan mencari masalah keamanan berdasarkan perspektif penyerang, selanjutnya merancang tindakan pencegahan. Prosesnya memungkinkan Product Owner dan Tim Pengembang memikirkan keamanan sebagai gambaran besar dari aplikasi atau fungsi yang selanjutnya bisa digunakan sebagai dasar membangun pertahanan, dan bukan hanya berfokus pada bug. Vektor klasik yang mencakup eskalasi kredensial pengguna yang tidak sah, keterbukaan informasi, penolakan (injeksi data palsu ke log), gangguan, spoofing, dan penolakan layanan. Untuk setiap fitur baru semua teknik subversi ini dievaluasi terhadap setiap tempat yang dikodekan satu pengguna atau kode modul yang lain. Jika masalah diidentifikasi, disempurnakan untuk mengatasi masalah. Di Agile, perubahan ini digabungkan ke dalam user stories sebelum dilakukan inkremen.

b. Daftar bug keamanan bisa dibuat dengan pelacakan bug keamanan mencakup pengumpulan data bug keamanan dan mendapatkan subset dari pengembang informasi yang perlu mengatasi masalah. Sebagian besar organisasi menyerahkan semua bug ditemukan hanya dengan sistem pelacakan bug. Bug dapat ditemukan dalam pengujian normal atau melalui tes keamanan yang dijelaskan di bawah ini. Alat pengujian keamanan memberi umpan pada sistem pelacakan bug sehingga masalah dapat dilacak, namun itu tidak berarti mereka memberikan informasi yang konsisten. Mereka juga tidak secara konsisten menilai kekritisan. Bagaimana mengintegrasikan dan menyesuaikan kekurangan pakan dari alat uji, dan menormalkan hasil tersebut, penting untuk integrasi Agile yang efektif. Perlu dicapai kesepakatan dengan Product Owner yang bugnya dapat ditangani setiap sprint berlangsung. Jaminan keamanan backlog harus ditinjau setiap sprint.

c. Manajemen patching dan konfigurasi pada sebagian besar perangkat lunak biasanya menggunakan kombinasi source code yang terbuka maupun komersial untuk melengkapi apa yang dibangun oleh Tim Pengembang. Misalnya Apache mendukung sebagian besar layanan web masa kini. Menjaga komponen sama pentingnya dengan memperbaiki masalah dalam kode. Agile menawarkan cara mudah untuk mengintegrasikan perubahan konfigurasi dan patch pada setiap awal sprint.

d. Metrik dan Manajemen Kebijakan: Penting untuk melacak tugas pengembangan, fitur, masalah yang terbuka dan rentan. Pentingnya mempunyai alat yang dapat memberikan pandangan tentang apa yang terjadi setiap hari atau setiap minggu diperlukan dalam mengontrol dan efisiensi proses pengembangan dan troubleshooting.

2.6.2 Sprint Backlog

Sprint Backlog adalah sekumpulan item Product Backlog yang telah dipilih untuk dikerjakan di Sprint, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan Sprint Goal. Sprint Backlog adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Inkremen selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang “Selesai”.

Sprint Backlog menampilkan semua pekerjaan yang dibutuhkan untuk mencapai Sprint Goal yang dibuat oleh Tim Pengembang.

Sprint Backlog adalah sebuah rencana yang cukup detail, di mana perubahan-perubahannya di tengah Sprint bisa dipahami saat Daily Scrum Meeting. Tim Pengembang memodifikasi Sprint Backlog sepanjang Sprint berlangsung, dan Sprint Backlog dapat berubah kapanpun juga sepanjang Sprint. Perubahan ini terjadi seiring dengan berkerjanya Tim Pengembang sesuai rencana pada saat itu, dan semakin meningkatnya wawasan tim untuk mencapai tujuan Sprint.

Dengan bertambahnya pekerjaan baru, Tim Pengembang menambahkannya ke dalam Sprint Backlog. Dengan dikerjakannya atau diselesaikannya pekerjaan, estimasi sisa pekerjaan juga diperbaharui. Ketika ada elemen dari perencanaan tidak dibutuhkan lagi, maka elemen tersebut dikeluarkan dari Sprint Backlog. Hanya Tim Pengembang yang dapat merubah Sprint Backlog pada saat Sprint sedang berjalan. Sprint Backlog sangat transparan, menggambarkan secara real-time pekerjaan yang akan diselesaikan oleh Tim Pengembang pada saat Sprint, dan ia sepenuhnya menjadi milik Tim Pengembang.

Memantau perkembangan Sprint

Di titik manapun dalam Sprint, jumlah sisa pekerjaan dalam Sprint Backlog dapat dijumlahkan. Tim Pengembang memantau sisa pekerjaan ini, setidaknya di setiap Daily Scrum, untuk memproyeksikan kemungkinan mereka akan mencapai Sprint Goal. Dengan memantau sisa pekerjaan ini sepanjang Sprint, Tim Pengembang dapat mengelola perkembangan pekerjaan.

2.6.3 Inkremen

Inkremen (tambahan potongan produk) adalah gabungan dari semua item Product Backlog yang diselesaikan pada Sprint berjalan dan nilai-nilai dari Inkremen sprint-sprint sebelumnya. Pada akhir Sprint, inkremen terbaru harus “Selesai”, yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi “Selesai” yang dibuat oleh Tim Scrum. Terlepas apakah Product Owner akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh.

2.7 Transparansi Artefak

Scrum berlandaskan transparansi. Keputusan-keputusan untuk mengoptimalisasi nilai produk dan mengendalikan resiko dibuat berdasarkan keadaan artefak hingga saat ini. Pada titik di mana transparansi berada pada tingkat tinggi, keputusan yang dibuat semakin dapat dipercaya. Pada titik di mana transparansi berada pada tingkat rendah, keputusan dapat dimanipulasi, nilai produk akan menurun dan resiko akan meningkat.

Scrum Master harus bekerja dengan Product Owner, Tim Pengembang dan pihak-pihak lain untuk bersama-sama memahami apakah semua artefak sudah sepenuhnya transparan. Ada banyak praktik untuk menangani belum penuhnya transparansi; dalam keadaan ini, Scrum Master harus membantu semua pihak untuk menerapkan praktik yang sesuai terhadap hilangnya transparansi. Scrum Master dapat mendeteksi hilangnya transparansi dengan meninjau semua artefak, mengamati pola-pola yang terjadi, menyimak apa yang dikatakan, dan melihat perbedaan antara apa yang diharapan dengan hasil yang sebenarnya.

Tugas Scrum Master adalah bekerja bersama Tim Scrum dan organisasi dimana ia berada untuk meningkatkan tingkat transparansi dari artefak-artefak yang digunakan. Pekerjaan ini biasanya membutuhkan pembelajaran, pendekatan persuasif, serta perubahan. Transparansi tidak terjadi dalam semalam, melainkan sebuah perjalanan jangka panjang.

2.8 Definisi Selesai

Ketika sebuah item Product Backlog atau Inkremen dikatakan “Selesai”, setiap pihak harus mengerti dengan apa yang dimaksud dengan “Selesai”. Walaupun definisi ini berbeda-beda antar tim Scrum, sesama anggota tim harus memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Ini adalah definisi selesai untuk Tim Scrum dan ini digunakan untuk memeriksa apakah pekerjaan untuk mengembangkan Inkremen dianggap selesai.

Definisi yang sama akan membimbing Tim Pengembang dalam mengetahui berapa banyak item Product Backlog yang mereka bisa ambil pada saat Sprint Planning. Tujuan dari setiap Sprint adalah untuk menghantarkan Inkremen, yang berpotensi untuk dirilis, yang memenuhi definisi “Selesai” terkini yang dibuat oleh Tim Scrum.

Tim Pengembang menghantarkan Inkremen yang berfungsi setiap Sprint. Inkremen ini dapat digunakan, supaya Product Owner dapat merilis produk tersebut sesegera mungkin jika ia mau. Apabila definisi “Selesai” untuk sebuah Inkremen adalah bagian dari konvensi, standar atau panduan pengembangan dari organisasi, setiap Tim Scrum harus mengikuti seluruhnya sebagai minimum requirement. Apabila “Selesai” untuk sebuah Inkremen bukan merupakan bagian dari konvensi, standar atau panduan pengembangan dari organisasi, Tim Pengembang harus membuat definisi “Selesai” yang pantas untuk produk yang dikembangkan. Apabila ada beberapa Tim Scrum yang mengembangkan sistem atau produk yang sama, seluruh Tim Pengembang di setiap Tim Scrum harus menentukan definisi selesai yang sama bersama-sama.

Setiap Inkremen merupakan gabungan dari Inkremen Sprint-sprint sebelumnya dan diuji secara teliti, untuk memastikan setiap Inkremen dapat berfungsi secara penuh.

Seiring dengan bertambah dewasanya Tim Scrum, mereka diharapkan untuk membuat definisi selesai yang lebih baik dan ketat lagi demi peningkatan kualitas. Produk atau sistem manapun harus memiliki definisi “Selesai” yang merupakan sebuah standar untuk pekerjaan yang akan dilakukan.

2.9 Security Acceptance Test

Beberapa contoh kategori story yang perlu dipertimbangkan yang memerlukan kriteria Security Acceptance antara lain:

2.9.1 Validasi Input

Developer perlu memahami pentingnya validasi input data apapun. Ketika sanitasi dan validasi data tidak dilakukan maka dapat menjadi sebab dari banyak jenis serangan. Buffer overflows, cross site scripting dan SQL injection hanyalah beberapa metode serangan yang sering dikenal karena gagal dalam validasi Input. Intinya ketika berhadapan dengan input data apapun, terutama yang berasal dari pengguna end-to-end, maka perlu divalidasi. Setiap story pengguna yang berhubungan dengan data Input perlu menyertakan kriteria penerimaan yang menyatakan bahwa data setidaknya perlu disanitasi dan divalidasi sesuai dengan yang diharapkan.

2.9.2 Penyimpanan atau Transfer Informasi Pribadi

Sebelum sebuah organisasi dapat menempatkan kontrol di sekitar informasi yang dibatasi, mereka terlebih dahulu harus menentukan apa yang dianggap pribadi oleh organisasi. Mulailah dengan mendefinisikan apa yang perlu dibatasi. Data mungkin termasuk dalam kategori seperti PII, PHI, IP, data pelanggan atau lainnya. Setiap story yang menyentuh kelas data ini perlu memiliki kriteria Security Acceptance. Bergantung pada bagaimana perusahaan memilih dalam melindungi, menyimpan, atau mentransmisikan data sehingga akan mendorong pengembangan story dan kriteria penerimaan.

2.9.3 Penggunaan Kriptografi

Ada beberapa aturan mendasar yang penting dalam kriptografi. Aturan pertama dan terpenting adalah tidak menggunakan algoritma kriptografi buatan sendiri. Aturan kedua adalah menggunakan penyedia library kriptografi yang terpercaya. Aturan ketiga adalah hanya menggunakan fungsi kriptografi yang disetujui sesuai dengan kebijakan keamanan perusahaan. Aturan keempat dan paling sulit adalah memastikan Anda menggunakan kriptografi untuk tujuan yang telah ditentukan. Pertahankan keempat aturan ini saat membuat sebuah story yang sama sekali berkaitan dengan kriptografi dan bagaimana hal itu dapat digunakan untuk mengembangkan kriteria penerimaan.

2.9.4 Otentikasi, Otorisasi, Akuntansi

Otentikasi, Otorisasi dan Akuntansi (AAA) adalah jantung dari setiap sistem kontrol akses. AAA adalah intelijen di balik pengendalian akses terhadap sumber daya, penegakan kebijakan dan audit kejadian tersebut. Mungkin lebih baik mengatakan bahwa segala sesuatu di AAA terkait dengan keamanan informasi. Saat mengembangkan sebuah story yang berhubungan dengan AAA, seseorang harus memastikan bahwa kriteria pengujian mencakup baik izin dan kasus penolakan untuk matriks akses keinginan. Misalnya, hanya menguji bahwa pengguna dapat masuk dengan kata kunci yang tepat tidak menganggap bahwa upaya kata kunci yang salah akan menolak akses. Hal umum lainnya yang perlu dipertimbangkan adalah bahwa tindakan AAA umumnya harus mencakup log event untuk tujuan audit.

2.9.5 Pengelolaan Log Keamanan

Setiap *software* aplikasi yang dikembangkan harus memiliki kemampuan *Logging Event* yang dapat digunakan untuk keperluan pengukuran dan pengawasan kinerja serta keamanannya. *Log software* aplikasi tersebut harus dikelola secara terpusat menggunakan *platform Security Incident & Event Management* (SIEM). *Alert* yang muncul sebagai keluaran dari SIEM harus didokumentasikan dan ditindaklanjuti oleh POM untuk menjadi *Item Product Backlog*.

DAFTAR PUSTAKA

http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ID.pdf

https://www.owasp.org/images/5/54/Owasp_stuttgart_agile_secure_20150803.pdf

https://advisera.com/27001academy/blog/2017/01/24/how-to-integrate-iso-27001-a-14-controls-into-the-system-software-development-life-cycle-sdlc/

https://www.slideshare.net/Cigital/agile-security-68073294

https://securosis.com/assets/library/reports/SecureAgileDevelopment_Nov2014_FINAL.pdf

https://devops.com/writing-security-acceptance-criteria-devops-story/