

Membuat Perencanaan Proyek untuk Manajer Pengembangan TIK

Tidak terasa kamu sudah menyelesaikan seluruh materi kelas “Membuat Perencanaan Proyek untuk Manajer Pengembangan TIK”. Untuk meninjau kembali materi yang telah kamu peroleh, baca PDF berikut yang menjelaskan rangkuman konten kelas.

Manajemen Proyek

“Bagaimana cara agar suatu proyek bisa dikelola dengan baik?”. Caranya adalah dengan mengoptimalkan manajemen proyek, yaitu metode yang digunakan untuk mengelola suatu proyek secara efektif dan efisien. Perencanaan adalah kunci sukses manajemen proyek. Semakin matang perencanaan maka potensi untuk meraih tujuan proyek akan semakin tinggi.

Proyek adalah serangkaian kegiatan yang bersifat unik dengan melibatkan stakeholder dan berlangsung dalam jangka waktu tertentu, serta memiliki indikator ketercapaian.

Contoh Proyek

Strategi Organisasi: Universitas Jaya Merdeka bertujuan menghasilkan akademisi dan profesional yang memiliki kompetensi unggul.

Portofolio: memfasilitasi mahasiswa dalam mengoptimalkan kegiatan akademik melalui pengadaan sarana dan prasarana

Program: program sarana di bidang IPTEK

Proyek: pembangunan toko komputer dan Wi-Fi yang bisa diakses 24 jam

Operasional: Pembukaan lowongan untuk penjaga toko.

Pengenalan Terhadap Agile

Agile merupakan sebuah *mindset* untuk dapat bertindak cepat, adaptif, dan fleksibel. *Agile* terdiri dari 4 *values* dan 12 *principles*. 4 *values agile* yaitu mengedepankan individu dan interaksinya dibandingkan proses dan *tools*, mengedepankan *software* yang berfungsi dibandingkan membuat dokumentasi lengkap, mengedepankan kerjasama dengan konsumen dibandingkan negosiasi kontrak, dan mengedepankan menanggapi perubahan dibandingkan sekedar mengikuti rencana. Sementara 12 *principles agile*, yaitu:

1. Prioritas utama adalah kepuasan konsumen melalui *software* yang dirilis dengan cepat dan rutin.
2. Terbuka dengan perubahan kebutuhan, bahkan di akhir pengembangan sekalipun. *Agile* mendukung perubahan ini untuk kebaikan konsumen.
3. Merilis *software* dengan rutin, mulai dari beberapa minggu, hingga beberapa bulan sekali. Akan tetapi, lebih cepat lebih baik.
4. *Developer* dan bagian bisnis harus bekerjasama setiap hari selama *project* berjalan.
5. Ciptakan *project* dengan individu yang penuh motivasi. Berikan mereka lingkungan dan *support* yang dibutuhkan. Lalu, percayalah mereka akan menyelesaikan pekerjaannya.
6. Metode paling efisien dan efektif untuk menyampaikan informasi di tim adalah secara empat mata.
7. *Software* yang berfungsi dengan baik merupakan tolok ukur kesuksesan *project*.
8. *Agile* mendukung pengembangan yang berkelanjutan sehingga semua pihak harus menjaga *pace* yang sama.
9. Aspek teknis dan desain yang bagus bisa meningkatkan kelincahan (*agility*).
10. Pentingnya kesederhanaan untuk meminimalisir pekerjaan yang tidak perlu.
11. Tim yang mandiri akan menghasilkan arsitektur, kebutuhan, dan desain terbaik.

12. Sesering mungkin, tim harus memikirkan cara untuk menjadi lebih efektif, lalu menerapkannya dengan baik.

Menjadi *Agile*

Menjadi *agile* adalah mereka yang peduli dengan jenis perilaku, pola pikir, dan pemikiran yang diperlukan untuk berhasil daripada berfokus pada *tools*, praktik, dan prosesnya. Orang yang menjadi *agile* memiliki beberapa karakteristik, yaitu mengedepankan *why* dalam mengembangkan produk, mendasarkan pada pertanyaan “mengapa saya harus lincah dan bagaimana?”, membutuhkan waktu yang lebih panjang, membutuhkan perubahan pola pikir, membutuhkan perubahan paradigma, melihat subjektif dan perspektif baru.

Sedangkan melakukan *agile* adalah mereka yang berfokus pada praktik atau kerangka kerja tanpa mempertimbangkan pola pikir *agile*. Sikap tersebut akan membahayakan proses transformasi *agile* karena tidak dilakukan dengan pola pikir yang benar. Orang yang melakukan *agile* memiliki beberapa karakteristik, yaitu mengedepankan *what* dalam mengembangkan produk, mendasarkan pada pertanyaan “apa yang harus saya lakukan agar bisa *agile*? Atau bagaimana cara terbaik agar bisa *agile*?”, membutuhkan waktu yang lebih singkat, bertentangan dengan sifat *agile* yakni lincah, dan mengedepankan kerangka kerja, metodologi, alat, dan teknik.

Fundamental dalam *Scrum*

Scrum merupakan *agile framework* yang terpopuler di dunia dan ditemukan oleh Jeff Sutherland dan Ken Schwaber. *Scrum* bukanlah sebuah proses, teknik, ataupun metodologi, namun sebuah kerangka kerja dimana dapat menggunakan bermacam proses dan teknik di dalamnya.

Pada tahun 2007, David Snowden menerbitkan artikel yang berjudul “A *Leader’s Framework for Decision Making*”. Isi artikel tersebut membagi empat kategori jenis permasalahan, yaitu *simple*, *complicated*, *complex*, dan *chaos*.

<i>Simple</i>	<i>Complicated</i>	<i>Complex</i>	<i>Chaos</i>
Sebab akibat yang sudah cukup jelas	Membutuhkan analisis dari <i>expert</i>	<i>Emergent answer</i>	Sebab dan akibat sukar diketahui
Pola yang berulang	Lebih banyak hal yang dapat diprediksi	Lebih banyak hal yang tidak dapat diprediksi	Terjadi banyak pergolakan yang membutuhkan keputusan
Manajemen berbasis fakta	Manajemen berbasis fakta	<i>Many competing ideas</i>	Tidak mempunyai cukup waktu untuk mengikuti prosedur bisnis <i>as usual</i>

Scrum memiliki 3 pilar, yaitu:

1. Transparansi, yang berarti semua pihak yang terlibat bisa bekerja bersama-sama.
2. Inspeksi, yang berarti bersikap kritis terhadap pekerjaan yang dilakukan.
3. Adaptasi, yang berarti mampu bereksperimen dengan peluang.

Di sisi lain, *scrum* juga memiliki 5 *value*, yaitu:

1. *Courage*, yang berarti memiliki keberanian untuk mengatasi masalah.
2. *Focus*, yang berarti *scrum* tim harus fokus dalam mencapai *sprint goals*.
3. *Commitment*, yang berarti berkomitmen untuk kesuksesan bersama.
4. *Solid*, yang berarti menjadi tim yang solid dan saling menghargai.
5. *Openness*, yang berarti membutuhkan transparansi dan keterbukaan.

Peran-Peran dalam Scrum



Gambar 4. Peran dalam Scrum

Peran-peran yang terdapat dalam *scrum* yaitu *product owner*, *scrum master*, dan *development team*. *Product owner* merupakan satu-satunya orang yang bertanggung jawab dalam pengelolaan *product backlog*. Contoh pengelolaan *product backlog* yaitu menyampaikan isi *product backlog* dengan jelas, mengurutkan *product backlog* item untuk mencapai tujuan, mengoptimalkan nilai bisnis dari pekerjaan *development team*, memastikan agar *product backlog* dapat dilihat, transparan, dan jelas, serta memastikan *development team* memahami *product backlog*.

Scrum master memiliki peran bertanggung jawab untuk memperkenalkan dan mendukung penggunaan *scrum* yang sesuai dengan *scrum guide*. Contoh pelayanan *scrum master* terhadap *development team* yaitu membimbing *development team* agar dapat menjadi tim yang *self-organized*, membantu *development team* untuk menghasilkan produk bernilai bisnis tinggi, menghilangkan hambatan yang memperlambat perkembangan pekerjaan *development team*, memfasilitasi acara-acara *scrum* bila diminta atau dibutuhkan, serta memberikan *coaching* kepada *development team* di organisasi dimana *scrum* belum sepenuhnya dipraktikkan dan dipahami.

Sedangkan contoh pelayanan *scrum master* terhadap organisasi yaitu menjadi *agile champion* bagi organisasi dalam penggunaan *scrum*, membuat perencanaan tentang implementasi *scrum* di dalam organisasi, membantu organisasi memahami dan menggunakan *scrum* dan pengembangan produk secara empiris, membuat perubahan yang dapat meningkatkan produktivitas *scrum team*, serta bekerja dengan *scrum master* lainnya untuk meningkatkan efektivitas dari penggunaan *scrum* di dalam organisasi.

Development team merupakan sebuah tim yang berjumlah tiga sampai sembilan orang dan berasal dari *cross-functional* tema. Mereka bekerja secara *self-organized* untuk menghantarkan *increment* yang siap dirilis setiap akhir *sprint*. *Development team* ini memiliki beberapa karakteristik, yaitu bekerja secara *self-organized*, anggota tim berasal dari *cross-functional team*, *scrum* tidak mengenal jabatan dalam *development team*, serta setiap *development team* bisa memiliki keahlian khusus tetapi akuntabilitasnya milik seluruh *development team*.

Dokumen dalam Scrum

Terdapat tiga artefak *scrum*, yaitu *product backlog*, *sprint backlog*, dan *increment*. Berikut ini perbedaan setiap artefak *scrum*nya:

<i>Product backlog</i>	<i>Sprint backlog</i>	<i>Increment</i>
<i>Product backlog</i> adalah daftar semua hal yang diketahui hingga saat ini harus ada dalam produk	<i>Sprint backlog</i> adalah hasil <i>sprint planning</i> dari daftar <i>product backlog item</i> yang terpilih	<i>Increment</i> adalah manifestasi dari <i>product backlog item</i>
<i>Product backlog</i> merupakan daftar dari seluruh fitur, fungsi, kebutuhan, peningkatan, dan perbaikan yang perlu diberlakukan terhadap	<i>Sprint backlog</i> haruslah transparan dan menampilkan semua aktivitas yang akan dikerjakan oleh <i>development team</i>	<i>Increment</i> merupakan tambahan untuk <i>increment</i> sebelumnya dan sudah dipastikan telah diuji secara seksama

produk pada rilis mendatang		
<i>Product backlog</i> memiliki atribut deskripsi, urutan, estimasi, dan nilai bisnis	<i>Sprint backlog</i> dapat dilihat secara jelas, menggambarkan keadaan terkini mengenai sisa pekerjaan yang telah direncanakan <i>development team</i>	<i>Increment</i> yang di <i>deliver</i> oleh <i>development team</i> harus sudah siap rilis
<i>Product backlog</i> adalah satu-satunya <i>requirement</i> yang akan digunakan untuk <i>development</i> produk	<i>Sprint backlog</i> merupakan hak milik <i>development team</i> sepenuhnya	<i>Increment</i> merupakan hasil pekerjaan yang bisa di inspeksi dan telah selesai guna mendukung empirisme di akhir <i>sprint</i>
<i>Product backlog</i> menjadi tanggung jawab <i>product owner</i>	<i>Sprint backlog</i> setidaknya ada satu proses <i>improvement</i> dengan prioritas tertinggi yang diambil dari hasil <i>outcome retrospective</i> pada <i>sprint</i> sebelumnya	
<i>Product backlog</i> merupakan <i>living artifact</i> , yaitu suatu artefak yang akan terus ada selama produknya masih <i>exist</i>		

Proses Pelaksanaan Scrum

Scrum event pertama adalah *sprint*. *Sprint* merupakan wadah dari semua *scrum event* yang harus melewati beberapa proses selama *sprint* berlangsung, yaitu *sprint planning*, *daily scrum*, *development*, *sprint review*, dan *sprint retrospective*. Hal-hal

yang perlu diperhatikan saat melakukan *sprint* yaitu tidak boleh ada perubahan *requirement* yang dapat mengancam *sprint goals*, kualitas produk tidak boleh menurun, *scope* dapat diklarifikasi dan dinegosiasi ulang antara *product owner* dan *development team* setiap kali adanya hal baru.

Sprint yang sedang berlangsung dapat dibatalkan dengan beberapa ketentuan, yaitu *sprint* dibatalkan sebelum *time box* berakhir, *sprint* hanya bisa dibatalkan oleh *product owner*, *sprint* bisa dibatalkan jika *sprint goals* berubah, ketika *sprint* dibatalkan maka *product backlog* ditinjau Kembali, serta pembatalan *sprint* memakan berbagai sumber daya.

Sprint planning merupakan perencanaan yang dilakukan oleh *scrum team* mengenai tujuan yang akan dicapai dalam sebuah *sprint* dan bagaimana cara mencapai tujuan tersebut. *Sprint planning* memiliki beberapa ketentuan, yaitu:

1. Perencanaan ini dilakukan secara kolaboratif oleh seluruh anggota *scrum team*.
2. *Sprint planning* memiliki batasan waktu maksimal delapan jam untuk *sprint* yang berdurasi satu bulan. Sementara untuk *sprint* yang lebih singkat acara ini biasanya lebih singkat.
3. *Scrum master* memastikan acara diselenggarakan dan peserta memahami tujuannya.
4. *Scrum master* mengedukasi *scrum team* untuk melakukan sesuai dengan *time box sprint planning*.
5. Pada saat *sprint planning* ada beberapa hal yang dapat dijadikan *input*, yaitu *product backlog*, kapasitas tim, dan *retrospektive action plan*.
6. *Outcome* dan *sprint planning* adalah *sprint goal* dan *sprint backlog*.

Daily scrum merupakan salah satu dari *scrum event* yang wajib dilakukan oleh *development team* setiap harinya dengan *time box* maksimal 15 menit. Terdapat tiga pertanyaan populer saat melakukan *daily scrum*, yaitu:

1. Apa saja pekerjaan kemarin yang dapat membantu *development team* mencapai *sprint goal*?

2. Apa yang akan saya kerjakan hari ini untuk membantu *development team* mencapai *sprint goal*?
3. Apakah saya melihat ada hambatan yang dapat mengganggu *development team* untuk mencapai *sprint goal*?

Sprint review merupakan *scrum event* yang dilakukan oleh *scrum team* dan *stakeholder* untuk menginspeksi *increments* dengan *time box* empat jam untuk *sprint* dengan durasi satu bulan. Hal-hal penting yang perlu diperhatikan saat *sprint review* yaitu:

1. *Sprint review* merupakan *event* wajib di dalam *scrum* meskipun ada *backlogs* yang belum selesai dihantarkan, *event* ini tetap harus dilaksanakan.
2. *Product owner* menyampaikan kepada *stakeholder* mengenai *backlog* yang belum selesai dan yang sudah selesai.
3. *Development team* mendemonstrasikan *increment* yang telah selesai, menjawab pertanyaan dari *stakeholder* dan menjelaskan tantangan selama mengerjakan *sprint backlog*.
4. Dilaksanakan untuk melihat kondisi *market* saat ini apakah cukup baik untuk merilis *increment*.
5. *Stakeholder* dan *scrum team* berkolaborasi untuk memberikan *input* pada *sprint* selanjutnya.

Sprint retrospective merupakan *event* terakhir yang dilakukan oleh *scrum team* setiap *sprint*. *Sprint retrospective* memiliki beberapa bertujuan, yaitu:

1. Menginspeksi *scrum team* dan membuat rencana peningkatan untuk dilaksanakan pada *sprint* berikutnya.
2. Menginspeksi jalannya *sprint* terakhir yang terkait dengan orang-orang, hubungan antar mereka, proses, dan alat-alat yang digunakan.
3. Mengidentifikasi dan mengurutkan hal utama yang berjalan dengan baik dan peningkatan yang berpotensi untuk dilakukan.
4. Membuat perencanaan untuk implementasi peningkatan cara kerja *scrum team*.