

AGILE Software Development

এজাইল সফটওয়্যার ডেভেলপমেন্ট

বাংলা টিউটোরিয়াল

By

Turzo Ahsan Sami

133014007

Dept. of Computer Science & Engineering

University of Liberal Arts Bangladesh

Submitted in partial fulfillment of the module

CSE 404 – Software Engineering

To

Nabeel Mohammed, PhD

সূচীপত্র

- ১। সফটওয়্যার ডেভেলপমেন্টের পদ্ধতি
- ২। এজাইল পদ্ধতি
- ৩। এজাইল মেনিফেস্টো
- ৪। এজাইল ডেভেলপমেন্টের প্রক্রিয়া
- ৫। এজাইল ফিলোসফি
- ৬। এজাইল টিম
- ৭। এজাইল ফ্রেমওয়ার্কঃ স্ক্রাম
- ৮। রেফারেন্স

১। সফটওয়্যার ডেভেলপমেন্ট পদ্ধতি

যেকোনও সফটওয়্যার বা সিস্টেম ডেভেলপ করার প্রধান শর্ত সফটওয়্যার বা সিস্টেমকে মার্কেটেবল প্রোডাক্ট হতে হবে। অর্থাৎ ক্লায়েন্ট বা ইউজার যেন সফটওয়্যার বা সিস্টেমকে ব্যবহার করে। কেবলমাত্র প্রোগ্রাম বা কোড লিখলেই একটি সফটওয়্যার তৈরী হয় না। সফটওয়্যার হতে হলে সেটাকে অবশ্যই ক্লায়েন্ট বা ইউজারের চাহিদা পূরণে সক্ষম হতে হবে। অর্থাৎ, সফটওয়্যার ডেভেলপমেন্টের মূলমন্ত্র হলঃ “ক্লায়েন্ট কী চায়?” এবং “সিস্টেমটা কী কাজ করবে?” ক্লায়েন্ট নিজে উল্লেখ না করে দিলে, সিস্টেমটা কীভাবে কাজ করবে বা কী টেকনোলজির ওপর নির্মিত হবে সেটা ডেভেলপমেন্ট টিমের ওপর বা সিস্টেম এনালিস্টের ওপর নির্ভর করে। ডেভেলপমেন্ট টেকনোলজি নির্ধারণ করা হয় সফটওয়্যার রিকয়ারমেন্ট, হার্ডওয়্যার রিকয়ারমেন্ট, ক্লায়েন্ট রিকয়ারমেন্ট, টার্গেট ইউজার গ্রুপ ইত্যাদির ওপর।

সফটওয়্যার ডেভেলপমেন্টের ক্ষেত্রে যেসব স্টেপ প্র্যাকটিস করা হয় সেগুলো হলঃ

১। রি-ইউজ টেকনোলজি

২। মেইনটেইনএবিলিটি

৩। স্ট্যাবিলিটি

৪। কোয়ালিটি

৫। স্কিল ট্রান্সফার

৬। ক্লায়েন্ট স্যাটিস্ফ্যাকশন

সফটওয়্যার ডেভেলপমেন্টের কোর প্রসেসগুলো হলঃ

১। রিকয়ারমেন্টস্

২। ডিজাইন

৩। কনসট্রাকশন

৪। টেসটিং

৫। ডিবাগিং

৬। ডিপ্লয়মেন্ট

৭। মেইনটেইনেন্স

সফটওয়্যার ডেভেলপ করার করার বেশ কিছু পদ্ধতি আছে। এই পদ্ধতিগুলো অবলম্বন করা হলে উপরে উল্লেখিত স্টেপ প্র্যাকটিস করা সহজতর হয়। সফটওয়্যার ডেভেলপমেন্টের পদ্ধতিগুলো মূলত একটি

ইনফর্মেশন সিস্টেম তৈরির সময়ে কাজের সঞ্চালন, পরিকল্পনা এবং গঠন প্রণালী নির্ণয়ে ব্যবহৃত হয়। সফটওয়্যার ডেভেলপমেন্ট পদ্ধতিগুলো প্যারাডাইম, মডেল, ফ্রেমওয়ার্ক এবং মেথডলজি নামে পরিচিত।

সফটওয়্যার ডেভেলপমেন্টের প্রধান প্যারাডাইম এবং মডেল সমূহ হলঃ

১। Software engineering

২। Waterfall

৩। Prototyping

৪। Incremental

৫। V-Model

৬। Dual Vee Model

৭। Spiral

৮। IID

৯। Agile

১০। Lean

১১। DevOps

সফটওয়্যার ডেভেলপমেন্টের প্রধান কিছু ফ্রেমওয়ার্ক এবং মেথডলজি সমূহ হলঃ

১। Cleanroom

২। TSP

৩। PSP

৪। RAD

৫। DSDM

৬। MSF

৭। Scrum

৮। Kanban

৯। UP

১০। XP

১১। TDD

১২। ATDD

১৩। BDD

১৪। FDD

১৫। DDD

১৬। MDD

এই পদ্ধতিগুলোর মধ্যে এস ডি এল সি, ইটারেটিভ ইনক্রিমেন্টাল মডেল, ওয়াটার ফল মডেল, এজাইল, ও স্ক্রাম বেশী জনপ্রিয়। ওয়াটার ফল পদ্ধতি হল ট্র্যাডিশনাল পদ্ধতি। এই পদ্ধতিতে সফটওয়্যার ডেভেলপমেন্ট ধাপে ধাপে সম্পন্ন হয়। পক্ষান্তরে এজাইল পদ্ধতিতে সফটওয়্যার ডেভেলপমেন্ট হয় সমান্তরালে। প্রকৃতপক্ষে এজাইল হল অনেকগুলো ছোট ছোট ওয়াটার ফল পদ্ধতির সমষ্টি। স্ক্রাম মূলত এজাইল পদ্ধতির একটি ফ্রেমওয়ার্ক। সফটওয়্যার ডেভেলপমেন্টের সময় বিভিন্ন ধরনের ঝুঁকি এড়াতে ওয়াটার ফল পদ্ধতি অপেক্ষা এজাইল পদ্ধতি বেশি জনপ্রিয়।

ওয়াটার ফল পদ্ধতিতে প্রথমেই রিকোয়ারমেন্টস সংগ্রহ করা হয়। এরপর এনালাইসিস ও ডিজাইন করা হয়। তারপর ইমপ্লিমেন্টেশন করা হয়। ইমপ্লিমেন্টেশন বলতে কেবলমাত্র ইন্টেলেশন থেকে শুরু করে লক্ষাধিক লাইনের কোড লেখাও হতে পারে। এরপর ডেভেলপড সিস্টেম টেস্ট করা হয়। এবং সবশেষে ক্লায়েন্টের কাছে ডেলিভার করা হয়। পুরো পদ্ধতিই ধাপে ধাপে সম্পন্ন হয়। সুতরাং, যদি রিকোয়ারমেন্টস সংগ্রহ করার সময়ই কোনও ভুল হয়ে যায় তবে ডেলিভারি করার পরই ভুল ধরা পড়বে, ফলে গুরুত্বপূর্ণ সময় ও রিসোর্স নষ্ট হবে। যদি রিকোয়ারমেন্টস ঠিক থাকে কিন্তু এনালাইসিস, ডিজাইন, ইমপ্লিমেন্টেশন বা কোডিং করার সময় ভুল হয় তাহলেও রিসোর্স অপচয় হবে। কারণ সিস্টেম ১০০ শতাংশ সঠিক না হলে ক্লায়েন্ট প্রোডাক্ট কিনবে না বা সার্ভিসের জন্য পে করবে না। সমীক্ষায় দেখা গেছে গড়ে ৩টি ডেভেলপড সিস্টেমের মধ্যে ২টিই প্রথমে রিজেক্টেড হয়। ফলে দেখা যাচ্ছে প্রায় ৬৬ শতাংশ প্রোডাক্ট আবার রিসোর্স খরচ করে টুইক বা টিউন করা লাগে। কোনও কোনও ক্ষেত্রে নতুন করে ডেভেলপ করা লাগে। এস ডি এল সি বা ওয়াটার ফল মডেলের এই ঝুঁকি এড়ানোর জন্য এজাইল পদ্ধতি বর্তমানে বেশি জনপ্রিয় হয়ে উঠেছে। পরিবেশ ও প্রজেক্টের ধরন অনুযায়ী অন্যান্য পদ্ধতিগুলোও ব্যবহৃত হয়।

২। এজাইল পদ্ধতিঃ

ডেভেলপমেন্ট রিস্ক মিনিমাইজ করার উদ্দেশ্যে এজাইল পদ্ধতির যাত্রা শুরু হয়। বিশ্বের অধিকাংশ ক্ষেত্রেই ক্লায়েন্ট একজন নন-টেকনিকাল পার্সোনেল হয়ে থাকেন। ফলে ক্লায়েন্টের রিকয়ারমেন্ট প্রথমবারেই সঠিকভাবে বুঝে ফেলে সেই অনুযায়ী ক্লায়েন্টকে স্যাটিসফাই করার মতন সফটওয়্যার সফলভাবে ডেভেলপ করার সম্ভাবনা ৩৩.৩৩ শতাংশ। ইন্ডাস্ট্রি বা এন্টারপ্রাইজ লেভেলের সফটওয়্যার ডেভেলপ করা ব্যয়বহুল। ইনফ্রাস্ট্রাকচার কস্ট, ইউটিলিটি কস্ট, ডেভেলপার ও ডেভেলপমেন্টের সাথে জড়িত পার্সোনেলদের স্যালারি ইত্যাদি সবই একটা সফটওয়্যার ডেভেলপমেন্টের কস্ট হিসেবে ধরা হয়। সেক্ষেত্রে যদি পুরো প্রোডাক্ট ডেভেলপ করা পর ক্লায়েন্ট রিজেক্ট করে দেয়, তবে পুরো প্রডাক্ট ডেভেলপকে অপচয় ধরা হয়। কারণ ঐ প্রডাক্টের আর কোনও উপযোগিতা নেই। এই অপচয় রোধ করার জন্য এজাইল পদ্ধতির উদ্ভব।

এজাইল পদ্ধতিতে প্রথমেই রিকয়ারমেন্টগুলোকে বেশ কিছু বিভাগে ভাগ করে ফেলা হয়। এই বিভাজনতা করা হয় মূলতঃ ডিপেন্ডেন্সি ক্যালকুলেট করে। ক্লাউড, মোবাইল, স্যোশাল টেকনোলজিস, ইন্টারনেট অফ থিংস, বিগ ডেটা ইত্যাদি প্রযুক্তির আগমনের সাথে সাথে ক্লায়েন্টের চাহিদাও বেড়ে চলেছে। ফলে ক্লায়েন্টের চাহিদার সাথে তাল মিলিয়ে চলার জন্য এবং ক্লায়েন্টের রিকয়ারমেন্টগুলো সঠিকভাবে ডেভেলপ করে ডেলিভার করার জন্য এজাইল সফটওয়্যার ডেভেলপমেন্ট বর্তমানে সফটওয়্যার ইন্ডাস্ট্রিতে ব্যাপক জনপ্রিয়তা লাভ করেছে। এজাইল পরিবর্তনের সাথে তাল মিলিয়ে চলতে সক্ষম একটা মেথডলজি।

বেশিরভাগ এজাইল ডেভেলপমেন্ট পদ্ধতি কোনও সমস্যা সমাধানের সময় এটিকে ছোট ছোট টাস্কে ভাগ করে নেয়। কোনও প্রকল্পের জন্যই সরাসরি কোনও দীর্ঘমেয়াদী পরিকল্পনা থাকে না। পুনরাবৃত্তিমূলক বিষয়গুলো স্বল্পমেয়াদী যেমন ১ থেকে ৪ সপ্তাহের মেয়াদে পরিকল্পনা করা হয়। প্রতিটা পুনরাবৃত্তিমূলক বিষয়ের জন্য একটি করে ক্রস ফাংশনাল টিম তৈরি করা হয় যা সফটওয়্যার ডেভেলপমেন্টের সকল ধাপ যেমন পরিকল্পনা, রিকোয়ারমেন্ট বিশ্লেষণ, ডিজাইন, কোডিং, ইউনিট টেস্টিং এবং এক্সপোস্টিং টেস্টিং করে থাকে। তাই প্রতিটা পুনরাবৃত্তিমূলক বিষয়ের ফলাফল হয় ওয়ার্কিং প্রোডাক্ট, এবং প্রতিটা পুনরাবৃত্তিমূলক ধাপের পর অংশীদারদের এর অবস্থান বুঝিয়ে দেয়া হয়। এর পরে রিভিউ কমেন্ট গ্রহণ করা হয় এবং কর্মক্ষম সফটওয়্যারের জন্য এর সংশ্লিষ্টতা যাচাই করে দেখা হয়। এজাইল মেথডলজি চিত্রের মাধ্যমে দেখানো হলঃ



৩। এজাইল মেনিফেস্টোঃ

২০০১ সালের ফেব্রুয়ারিতে ইউটাহ এর স্নোবোর্ড রিসোর্টে ১৭ জন স্বাধীন-মনস্ক সফটওয়্যার প্র্যাকটিশনার একটি কনফারেন্সের আয়োজন করেন লাইটওয়েট ডেভেলপমেন্ট নিয়ে আলোচনা করার জন্য। এই কনফারেন্স এর ফলাফল হল সফটওয়্যার ডেভেলপমেন্টের এজাইল মেনিফেস্টো। এজাইল মেনিফেস্টোর ৪টি মূলনীতি হলঃ

১। Individuals and interactions over processes and tools- ব্যক্তি ও ব্যক্তিকেন্দ্রিক মিথস্ক্রিয়াকে প্রক্রিয়া ও যন্ত্রপাতির উর্ধ্বে মূল্যায়ন করা

২। Working software over comprehensive documentation- কার্যরত সফটওয়্যারকে কম্প্রিহেনসিভ ডকুমেন্টেশনের উর্ধ্বে মূল্যায়ন করা

৩। Customer collaboration over contract negotiation - কাস্টোমার সহযোগিতাকে চুক্তিপত্রের আলাপআলোচনার উর্ধ্বে মূল্যায়ন করা

৪। Responding to change over following a plan - পরিবর্তনশীল আচরণের সাথে তাল মিলিয়ে চলাকে পরিকল্পনা অনুসরণের উর্ধ্বে মূল্যায়ন করা

অর্থাৎ মূলউপজীব্য বিষয় হলঃ

১। ব্যক্তি ও মিথস্ক্রিয়া

২। কার্যরত সফটওয়্যার

৩। কাস্টোমার সহযোগিতা

৪। পরিবর্তনের সাথে তাল মিলিয়ে চলা

এজাইল মেনিফেস্টোর ১২ টি মূলনীতি হল:

- ১। ক্লায়েন্ট স্যাটিসফ্যাকশন: সচল ও দ্রুত ডেভেলপমেন্টের মাধ্যমে ক্লায়েন্ট স্যাটিসফ্যাকশন অর্জন করা। যেহেতু সফটওয়্যারের মূল এভালুয়েটর হচ্ছে ক্লায়েন্ট, তাই দ্রুত এবং ক্রমাগত সফটওয়্যার বা সফটওয়্যারের ব্যবহারযোগ্য অংশ ডেলিভার করতে হবে।
- ২। পরিবর্তনকে স্বাগত জানানো: সফটওয়্যার ডেভেলপমেন্টে পরিবর্তন অবশ্যম্ভাবী, তাই পরিবর্তনকে স্বাগত জানানো একটি প্রধান বিষয়। কারণ একেবারে শেষ স্টেপেও ক্লায়েন্ট নিজের বাণিজ্যিক স্বার্থে রিকয়ারমেন্ট পরিবর্তিত হতে পারে, তাই ক্লায়েন্ট স্যাটিসফ্যাকশনের লক্ষ্যে পরিবর্তনকেই অবশ্যকর্তব্য মনে করতে হবে।
- ৩। স্বল্প সময়ে যেমন কয়েক সপ্তাহ থেকে কয়েক মাসের মধ্যে ক্লায়েন্টের চাহিদা অনুযায়ী একটি কার্যকর সফটওয়্যার ডেলিভার করতে হবে, যতদ্রুত সম্ভব। ছোট টাইমলাইনকে অগ্রাধিকার দিতে হবে।
- ৪। সমন্বয় সাধন: ব্যবসায়িক লোকজন ও ডেভেলপারদের প্রকল্পের পুরোটা সময় সমন্বয়ের মাধ্যমে কাজ করতে হবে।
- ৫। মোটিভেশন: ব্যক্তি পর্যায়ে মোটিভেট করার প্রবণতা নিয়ে প্রকল্প গ্রহণ করতে হবে।
- ৬। মুখোমুখি কথোপকথন: ডেভেলপমেন্ট টিমে মুখোমুখি কথোপকথন সবচেয়ে কার্যকর ও গঠনমূলক যোগাযোগ প্রক্রিয়া।
- ৭। কার্যরত সফটওয়্যার অনুযায়ী কাজের অগ্রগতি পরিমাপ করা: কার্যরত সফটওয়্যার একটি মূল বিষয়। এর মাধ্যমে অগ্রগতি পরিমাপ করা যাবে।
- ৮। এজাইল সাসটেইনেবল ডেভেলপমেন্টের জন্য কন্সটেন্ট প্যাক মেইনটেইন করবে।
- ৯। মনিটরিং: গতি বৃদ্ধির জন্য টেকনিক্যাল বিষয় ও ভাল ডিজাইনের জন্য মনিটরিং করা
- ১০। সারল্য: বিষয়বস্তুগুলো সহজ রাখা ও সরল টার্ম ব্যবহার করা
- ১১। স্বনিয়ন্ত্রিত টিম: আদর্শ এজাইল টিম স্বনিয়ন্ত্রিত হবে, অন্য টিমের উপর বেশি নির্ভর করবে না কারণ স্বতন্ত্র টিম থেকেই সেরা স্থাপত্যগত, ডিজাইনভিত্তিক ও প্রত্যাশামত সফটওয়্যার পাওয়া যায়।
- ১২। নিয়মিত কাজ রিভিউ করা: নিয়মিত কাজ রিভিউ করলে টিম জানতে পারে কীভাবে আরও বেশি দক্ষ হওয়া যায় ও সেই অনুযায়ী বিবেচনার সমন্বয় করতে পারে।

৪। এজাইল ডেভেলপমেন্টের বিভিন্ন প্রসেস

ইউজার স্টোরি:

ইউজার স্টোরি হল এক ধরনের রিকয়ারমেন্ট যা ইউজারের সহজ ভাষায় কয়েক লাইনে বিবৃত হয়। ইউজার স্টোরি একটি ইটারেটিভ স্টেপে সম্পন্ন হয়। একটি ইউজার স্টোরি তখনই সম্পন্ন হবে যখন নিচের কাজগুলো সম্পন্ন হবে:

- ১। সংশ্লিষ্ট সব কোড অন্তর্ভুক্ত হয়েছে
- ২। সব ইউনিট টেস্ট কেস পার হয়েছে
- ৩। সব এক্সপটেশন টেস্ট কেস পার হয়েছে
- ৪। হেল্প টেক্সট লেখা হয়েছে
- ৫। প্রোডাক্ট ওনার স্টোরি গ্রহণ করেছে

ইটারেটিভ স্টেপ বা স্প্রিন্ট:

ইটারেটিভ স্টেপ হল নির্দিষ্ট সময়ের ফ্রেমে ইউজার স্টোরির কালেকশন, সমস্যা ও গ্রহণযোগ্যতার সমন্বয় যা উৎপাদনের খেত্রে বিবেচিত হবে। এই ইটারেটিভ স্টেপগুলো বিবৃত হয় তখন, যখন ইটারেটিভ স্টেপের পরিকল্পনা, খসড়া ও রিভিউ মিটিং করা হয়। একে স্প্রিন্টও বলা হয়। একটি ইটারেটিভ স্টেপ সম্পন্ন হয় তখন যখন নিচের কাজগুলো সম্পন্ন হয়:

- ১। প্রোডাক্ট ব্যাকআপ সম্পন্ন
- ২। পারফরমেন্স পরীক্ষা
- ৩। ইউজার স্টোরিগুলো গৃহীত ও পরবর্তী স্টেপে পাঠানো
- ৪। এরর সারানো অথবা পরবর্তী স্টেপের জন্য স্থগিত করা হয়েছে

রিলিজ:

রিলিজ একটি বড় মাইলস্টোন যা প্রকল্পের আভ্যন্তরীণ বা বাইরে উৎপাদন, টেস্টকৃত ভার্সন ইত্যাদি বোঝায়। একটি রিলিজ তখনই সম্পন্ন হবে নিচের স্টেপগুলো সম্পন্ন হয়:

- ১। সিস্টেম স্ট্রেস টেস্ট
- ২। পারফরমেন্স যাচাই
- ৩। সিকিউরিটি ভ্যালিডেশন
- ৪। রিস্ক ম্যানেজমেন্ট পরিকল্পনা যাচাই

সরাসরি যোগাযোগ:

প্রত্যেকটি এজাইল টিমের একজন করে ক্লায়েন্ট প্রতিনিধি(যেমন,পণ্যের মালিক)থাকেন, তিনি অংশীদারদের প্রতিনিধি, এবং প্রতিটা পুনরাবৃত্তিমূলক ধাপের পর ডেভলপারদের অনুসন্ধানমূলক প্রশ্নের উত্তরও দিয়ে থাকেন। একটি ইনফরমেশন রেডিয়েটর যা কিনা একটি ফিজিকাল ডিসপ্লে, অফিসে সুস্থিতভাবে বসানো থাকে, যার মাধ্যমে যে কেউ এজাইল টিমের অগ্রগতি সম্পর্কে জানতে পারে। এই ইনফরমেশন রেডিয়েটর প্রকল্পের সর্বশেষ অবস্থা নির্দেশ করে থাকে।

ফিডব্যাক লুপঃ

ডেইলি স্ট্যান্ডআপ এজাইল ডেভলাপমেন্টের একটি কমন ধারা, এটি ডেইলি স্কাম নামেও পরিচিত। এখানে প্রত্যেক সদস্য সংক্ষিপ্ত সেশনে নিজেদের সম্পন্ন করা কাজের অগ্রগতি, ভবিষ্যৎ ও এক্ষেত্রে কিসের মোকাবেলা করতে হচ্ছে তা নিয়ে আলোচনা করে।

প্রোডাক্ট ব্যাকলগঃ

প্রোডাক্ট ব্যাকলগ হচ্ছে সম্পাদন করা হবে এমন একটি তালিকা। আইটেমগুলোকে বৈশিষ্ট্য অনুযায়ী স্থান দেওয়া হয়। আদর্শ ক্ষেত্রে আইটেমগুলোকে ইউজার স্টোরি অনুযায়ী বিভাজন/Divide করা উচিত। প্রত্যেক প্রোডাক্টের একটি করে প্রোডাক্ট ব্যাকলগ থাকা উচিত যার বড় এবং আরো বড় বৈশিষ্ট্যের সেট থাকতে পারে। একাধিক দল একটি ব্যাকলগ এর উপর কাজ করতে পারে। ব্যবসায়িক মান, প্রযুক্তিগত মান, ঝুঁকি ব্যবস্থাপনা বা কৌশলগত ফিটনেস বৈশিষ্ট্যের উপর ভিত্তি করে পণ্যের র‍্যাঙ্ক নির্ণয় করা হয়। উচ্চ র‍্যাংকিং সম্পন্ন পণ্য রিলিজ পরিকল্পনা করার সময়ের ছোট ছোট ঘটনার উপর ভিত্তি করে এই পণ্যের পুনরাবৃত্তি করা হবে কিনা তা নির্ধারণ করে।

প্রোডাক্ট ব্যাকলগ গুরুত্বপূর্ণ কেন?

১। এটি তৈরি করা হয় যাতে প্রত্যেকটি এবং সকল বৈশিষ্ট্য এর আনুমানিক হিসাব করা যায়।

২। এটি পণ্যের রোডম্যাপ পরিকল্পনায় সাহায্য করে।

৩। এটা বৈশিষ্ট্যসমূহের পুণরায় প্রাধান্য নির্ধারণ করতে সাহায্য করে, যাতে পণ্যের মান আরো বাড়ানো যায়।

৪। এটি কোন বৈশিষ্ট্যটি প্রথম প্রাধান্য পাবে তা নির্ণয় করতে সাহায্য করে। সকল বৈশিষ্ট্য এর একসাথে র‍্যাংক নির্ণয় করে এবং এরপরে মূল্য নির্ধারণ করে।

ডেইলি স্ট্যান্ডআপঃ

এজাইল পরিবর্তনের সাথে তাল মিলিয়ে চলতে সক্ষম একটা মেথড। ডেইলি স্ট্যান্ডআপ হল এজাইল টিমের সকল সদস্যদের মধ্যে দৈনিক স্ট্যাটাস মীটিং। এটি কাজের রেগুলার আপডেটই কেবল দেয় না বরং সব সদস্যের সমস্যাগুলোকে সামনে আনে যার ফলে এগুলো সমাধান করা সহজ হয়। ডেইলি স্ট্যান্ডআপ এজাইল টিমের একটি অবশ্য করণীয় কাজ, এবং সবসময় এর চর্চা দরকার, তা যেভাবেই এজাইল টিমের গঠন হয়ে থাক না কেন। এটি মূলত এজাইল টিমের সকল সদস্যদের মধ্যে দৈনিক স্ট্যাটাস মীটিং আর সাধারণত ১৫ মিনিট স্থায়ী হয়। প্রত্যেক সদস্যকে তিনটি গুরুত্বপূর্ণ প্রশ্নের উত্তর দিতে হয় যেগুলো হলঃ গতকাল কী করেছি? আগামীকাল কী করছি? এবং কোনও বাধা বিপত্তি আছে কী না? ডেইলি স্ট্যান্ডআপ মূলত স্ট্যাটাস আপডেটের জন্য, এতে আলোচনা তেমন হয় না।

আলোচনার জন্য ভিন্ন সময়ে ভিন্ন ধরনের মিটিংয়ের ব্যবস্থা করতে হয়। ডেইলি স্ট্যান্ডআপে সাধারণত অংশগ্রহণকারীদের দাঁড়িয়ে কথা বলতে হয়, এজন্য মোটামুটি দ্রুতই এই মিটিং শেষ করা হয়।

ডেইলি স্ট্যান্ডআপ কেন গুরুত্বপূর্ণঃ

১। টিম দৈনন্দিন অগ্রযাত্রা সম্পর্কে অবহিত থাকে, পর্যায়বৃত্তিক ধাপের প্ল্যানমত কাজ এগুচ্ছে কিনা তা জানা যায়

২। প্রত্যেক সদস্য তার লক্ষ্য সম্পর্কে বাকিদের জানায়

৩। কোনও বিলম্ব বা বাধা থাকলে তা দৃশ্যমান হয়

ডেইলি স্ট্যান্ডআপে দৈনিক ভিত্তিতে উপস্থিত থাকেন স্ক্রাম মাস্টার, প্রোডাক্ট ওনার, ডেলিভারি টিম।

পার্টনার, ফোকাসগ্রুপ ও ক্লায়েন্টদের উপস্থিত থাকতে উৎসাহিত করা হয়। প্রত্যেক সদস্যের অনুসন্ধান ও সমস্যার বিষয়ে নোট নেয়ার দায়িত্ব স্ক্রাম মাস্টারের। ভৌগলিকভাবে দূরে অবস্থিত টিমের

ক্ষেত্রে ডেইলি স্ট্যান্ডআপ বিভিন্নভাবে হতে পারে, ভিন্ন টাইম জোনে অবস্থিত সদস্যদের জন্য যা করণীয়ঃ

১। রোটেশনভিত্তিক একজন সদস্য নির্বাচন করতে হবে যে ভিন্ন টাইম জোনে অবস্থিত সদস্যদের নিয়ে মিটিং করতে পারবে

২। প্রতি টিমের জন্য ভিন্ন স্ট্যান্ডআপ মিটিংয়ের আয়োজন করা এবং রেলি, ট্রেলো, শেয়ারপয়েন্টের মত মাধ্যমে স্ট্যাটাস আপডেট রাখা

৩। বিভিন্ন ধরনের কমুনিকেশন টুল প্রস্তুত রাখা

এজাইল প্র্যাকটিসঃ

এজাইল ডেভেলপমেন্ট কিছু কংক্রিট প্র্যাকটিসের ওপর স্থাপিত। এজাইল ডেভেলপমেন্টে সেসব প্র্যাকটিস দ্বারা রিকয়ারমেন্ট, ডিজাইন, মডেলিং, কোডিং, টেস্টিং, প্ল্যানিং, রিস্ক ম্যানেজমেন্ট, প্রসেস, কোয়ালিটি ইত্যাদি নিরূপণ করা সম্ভব। প্র্যাকটিসগুলো হলঃ

১। এক্সপটেন্স টেস্ট ড্রিভেন ডেভেলপমেন্ট (ATDD)

২। এজাইল মডেলিং

৩। প্রডাক্ট ব্যাকলগ এবং স্প্রিন্ট

৪। বিহেভিয়ার ড্রিভেন ডেভেলপমেন্ট (BDD)

৫। বিজনেস এনালিস্ট ডিজাইনার মেথড (BADM)

৬। ক্রস ফাংশনাল টিম

৭। কন্টিনিউয়াস ইন্টগ্রেশন (CI)

৮। ডোমেইন ড্রিভেন ডেভেলপমেন্ট (DDD)

৯। ইনফরমেশন রেডিয়েটরস -স্ক্রাম বোর্ড, টাস্ক বোর্ড, ভিজুয়াল ম্যানেজমেন্ট বোর্ড, বার্নডাউন চার্ট

১০। ইটারেটিভ ও ইনক্রিমেন্টাল ডেভেলপমেন্টাল (IID)

১১। পেয়ার প্রোগ্রামিং

১২। প্ল্যানিং পোকার

১৩। রিফ্যাকটরিং

১৪। স্ক্রাম ইভেন্টস- স্প্রিন্ট প্ল্যানিং, ডেইলি স্ক্রাম, স্প্রিন্ট রিভিউ এবং রেট্রোস্পেক্টিভ

১৫। টেস্ট-ড্রিভেন ডেভেলপমেন্ট (TDD)

১৬। এজাইল টেস্টিং

১৭। টাইমবক্সিং

১৮। ইউজার স্টোরি

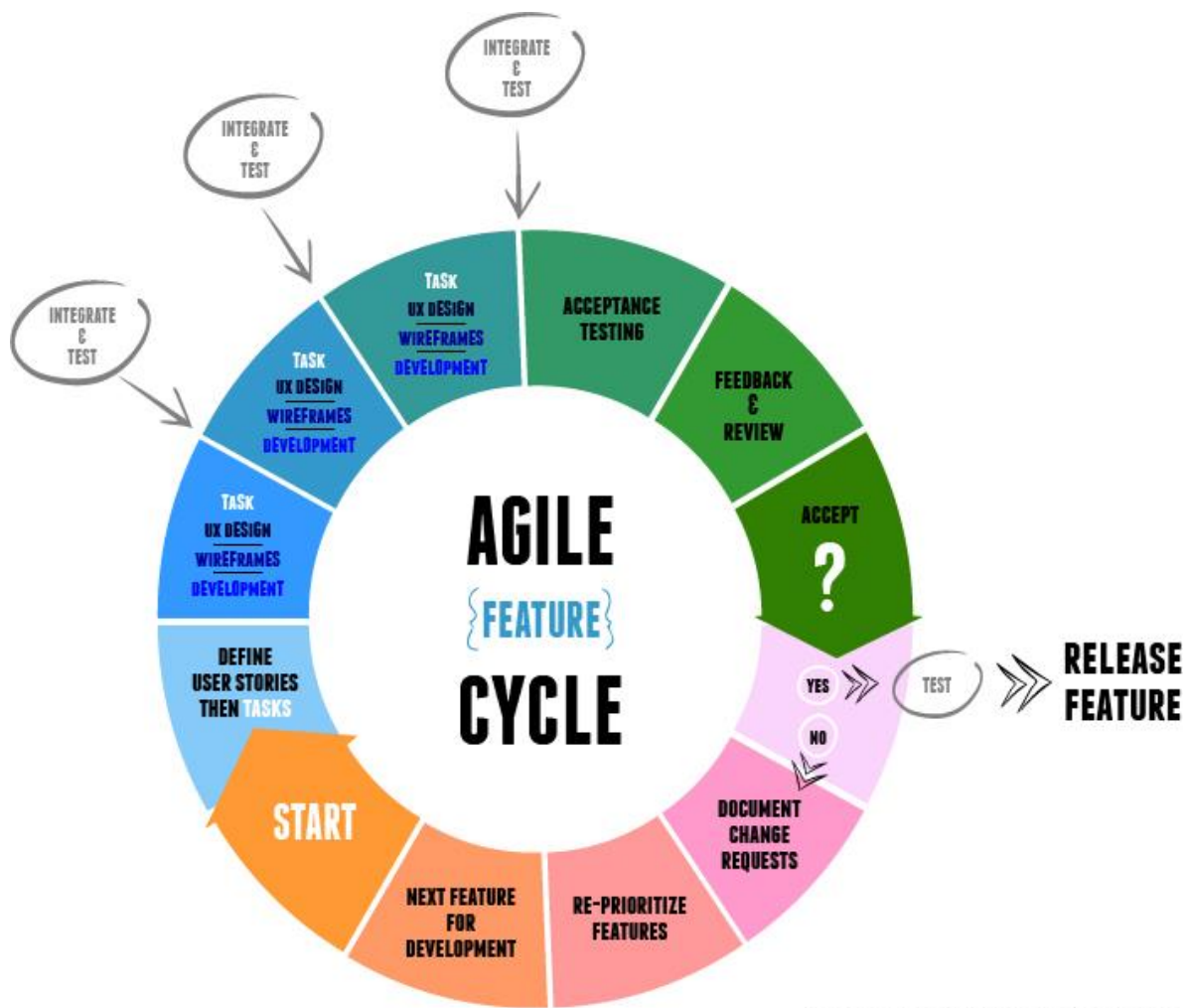
১৯। স্টোরি ড্রিভেন মডেলিং

২০। রেট্রোস্পেক্টিভ

২১। ভেলোসিটি ট্র্যাকিং

২২। ইউজার স্টোরি ম্যাপিং

এজাইলের প্রক্রিয়া চিত্রের সাহায্যে দেখানো হলঃ



Agile Feature Cycle by ChromeMedia Inc.

৫। এজাইল ফিলোসফি

প্রথাগত সফটওয়্যার ডেভেলপমেন্টের চেয়ে এজাইল ডেভেলপমেন্ট স্বতন্ত্র কারণ এজাইল ডেভেলপমেন্ট মূলত টার্গেট করে কমপ্লেক্স সিস্টেম ও প্রোডাক্ট এমনভাবে ডেভেলপ করতে হবে যেন সেটা ডাইনামিক, নন-ডিটারমিনিস্টিক ও নন-লিনিয়ার হয়। সফটওয়্যার ডেভেলপমেন্টে প্রায়ই স্টেবল পরিকল্পনা গুরু স্টেপগুলোতে অনুপস্থিত থাকে। একইসাথে অনুমানভিত্তিক পরিকল্পনাও অধিকাংশ সময় ভুল প্রমাণিত হয়। ফলে অপচয়ের সম্ভাবনা বাড়তে থাকে। এই সমস্যা দূর করার জন্য কিংবা হ্রাস করার জন্য এজাইল মেথড খুবই কার্যকরী। বহু বছর ধরে ইন্ডাস্ট্রির সাফল্য ও ব্যর্থতা অবজার্ব করার পর এজাইল মেথড ডেভেলপ করা হয়েছে। এজাইল মেথড এডাপটিভ, ইটারেটিভ ও এভোলিউশনারি- ফলে, সাসটেইনেবল।

এডাপটিভ বনাম প্রেডিকটিভঃ

সফটওয়্যার ডেভেলপমেন্ট মেথড এডাপটিভও হতে পারে, আবার প্রেডিকটিভও হতে পারে। এজাইল মেথড এডাপটিভ। এডাপটিভ ডেভেলপমেন্টের মূলমন্ত্র হল সমুদ্রের ঢেউ এর মতন পরিবর্তনশীল লক্ষ্যমাত্রা অর্জন। এডাপটিভ মেথড অনুযায়ী, লক্ষ্য জানা থাকতে হবে, অর্থাৎ ক্লায়েন্ট কী চাচ্ছে বা সফটওয়্যার কী করবে সেটা পরিষ্কারভাবে জানতে হবে এবং ডেভেলপ করতে হবে কিন্তু কীভাবে সেটা অর্জন করতে হবে সেই পরিকল্পনা তাৎক্ষণিক অবস্থা অনুযায়ী পরিবর্তনের ক্ষমতা থাকতে হবে। অর্থাৎ গন্তব্য স্থির, কিন্তু পথটা ফ্লেক্সিবল। এডাপটিভ টিম ৭ দিন পর কী কাজ করবে সেটা অনুমান করতে অক্ষম কিন্তু এই মুহুর্তে ক্লায়েন্ট কী কী ফিচার চাচ্ছে সেটা পরিষ্কারভাবে বলতে ও ডেভেলপ করতে সক্ষম। ক্লায়েন্টের চাহিদা বদলের সাথে এডাপটিভ টিমের কাজও বদলাতে থাকে।

কোড বনাম ডকুমেন্টেশনঃ

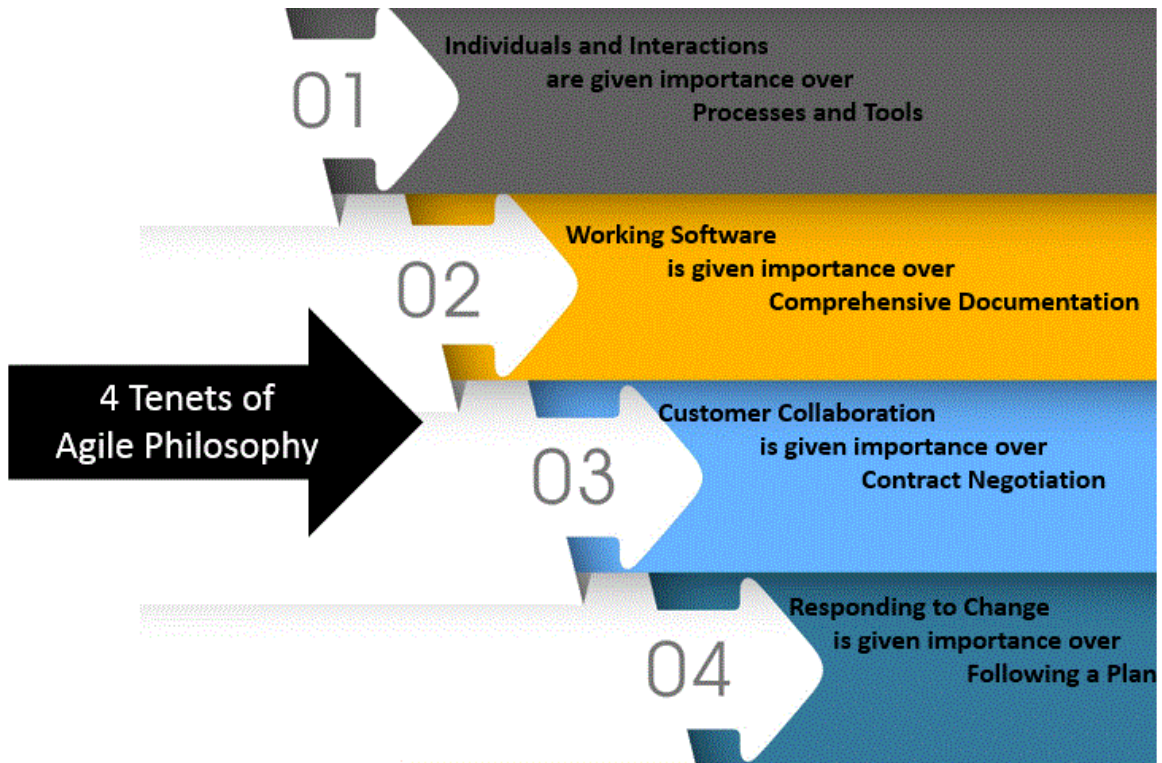
যদিও সফটওয়্যার ডেভেলপমেন্টের একটা গুরুত্বপূর্ণ অংশ হল ডকুমেন্টেশন, সফটওয়্যার ডেভেলপাররা ডকুমেন্টেশন লেখার চাইতে কোড লিখতে বেশি স্বাচ্ছন্দ্য বোধ করেন। এজাইল ফিলোসফি অনুযায়ী, যদি ক্লায়েন্টের বেঁধে দেয়া লক্ষ্যমাত্রা অর্জনের জন্য যদি ডকুমেন্টেশন লেখাটাই সেরা উপায় হয়, তবে ডকুমেন্টেশন লিখতেই হবে। কিন্তু যদি লক্ষ্যমাত্রা অর্জনের জন্য ডকুমেন্টেশন লেখার চাইতে ভাল উপায় থাকে তবে সেই উপায়ই অবলম্বন করা উচিত। কারণ এজাইল অনুযায়ী, ক্লায়েন্টের লক্ষ্য অর্জনই ডেভেলপমেন্ট টিমে একমাত্র কাজ। তাই যদি ডকুমেন্টেশন লেখা অপচয় হিসেবে চিহ্নিত হয় তবে খুব বেশি নিখুত ডকুমেন্টেশন লেখার দরকার নেই, বরং কোড অনেক সময়ই ডকুমেন্টের চাইতে আলাদা হয়- এমনটাই হচ্ছে এজাইল মেথডের ফিলসফি। অর্থাৎ সম্ভাব্য সেরা উপায়ে কার্যকর সফটওয়্যার ডেভেলপ করাটাই এখানে মূল উপজীব্য। তবে মেইনটেইনেন্সের জন্য কিছু মাত্রার ডকুমেন্টেশন তো লিখতেই হবে।

ইটারেটিভ বনাম ওয়াটারফলঃ

ওয়াটার ফল পদ্ধতিতে প্রথমেই রিকোয়ারমেন্টস সংগ্রহ করা হয়। এরপর এনালিসিস ও ডিজাইন করা হয়। তারপর ইমপ্লিমেন্টেশন করা হয়। ইমপ্লিমেন্টেশন বলতে কেবলমাত্র ইন্সটলেশন থেকে শুরু করে লক্ষ্যধিক লাইনের কোড লেখাও হতে পারে। এরপর ডেভেলপড সিস্টেম টেস্ট করা হয়। এবং সবশেষে ক্লায়েন্টের কাছে ডেলিভার করা হয়। পুরো পদ্ধতিই ধাপে ধাপে সম্পন্ন হয়। সুতরাং, যদি রিকোয়ারমেন্টস

সংগ্রহ করার সময়ই কোনও ভুল হয়ে যায় তবে ডেলিভারি করার পরই ভুল ধরা পড়বে, ফলে গুরুত্বপূর্ণ সময় ও রিসোর্স নষ্ট হবে। যদি রিকোয়ারমেন্টস ঠিক থাকে কিন্তু এনালাইসিস, ডিজাইন, ইমপ্লিমেন্টেশন বা কোডিং করার সময় ভুল হয় তাহলেও রিসোর্স অপচয় হবে। কারণ সিস্টেম ১০০ শতাংশ সঠিক না হলে ক্লায়েন্ট প্রোডাক্ট কিনবে না বা সার্ভিসের জন্য পে করবে না। সমীক্ষায় দেখা গেছে গড়ে ৩টি ডেভেলপমেন্ট সিস্টেমের মধ্যে ২টিই প্রথমে রিজেক্টেড হয়। ফলে দেখা যাচ্ছে প্রায় ৬৬ শতাংশ প্রোডাক্ট আবার রিসোর্স খরচ করে টুইক বা টিউন করা লাগে। কোনও কোনও ক্ষেত্রে নতুন করে ডেভেলপ করা লাগে। একারণে এজাইল মেথডে রিকোয়ারমেন্টসগুলোকে কয়েকটা ভাগে ভাগ করে নিয়ে কয়েক ইটারেশনে ছোট ছোট ওয়াটার ফল মডেলে ডেভেলপ করা হয়। ফলে অপচয়ের পরিমাণ অনেকাংশে কমে আসে, কারণ প্রথমে একটা বিভাগ ক্লায়েন্ট রিজেক্ট করলে সেটা শোধরানোর মতন যথেষ্ট সময় পাওয়া যায়। ইটারেটিভ মডেল অনেকটা কম্পিউটার আর্কিটেকচারের পাইপলাইনের সাথে তুলনা করা যায়। প্রথম বিভাগের প্রথম স্টেপ শেষ হয়ে যখন দ্বিতীয় স্টেপে প্রবেশ করে তখন দ্বিতীয় বিভাগের প্রথম স্টেপ শুরু হয় এবং এভাবেই ডেভেলপমেন্ট শেষ হয়। অর্থাৎ ক্লায়েন্ট কয়েকটা ভাগে বিভক্ত অথচ কার্যকর সফটওয়্যার এভালুয়েট করে।

নিচের চিত্রের সাহায্যে এজাইল ফিলোসফি দেখানো হল:



৬। এজাইল টিম

এজাইল টিমে সাধারণত ৫-৯ জন পার্সোনেল থাকেন। এজাইল টিম ক্ষমতাসম্পন্ন, ম্যানেজড ও দায়িত্ববোধসম্পন্ন। এজাইল টিম একসাথে প্লান করে, একসাথে ডিজাইন করে, একসাথে ইমপ্লিমেন্ট করে এবং একসাথে শেখে।

এজাইল টিমের রেসপন্সিবিলিটিগুলো হলঃ

১। এজাইল টিম নিজেদের কাজ ম্যানেজ করতে সক্ষম

২। এজাইল টিম কাজের কমপ্লেক্সিটি ও সাইজ বিশ্লেষণ করতে সক্ষম

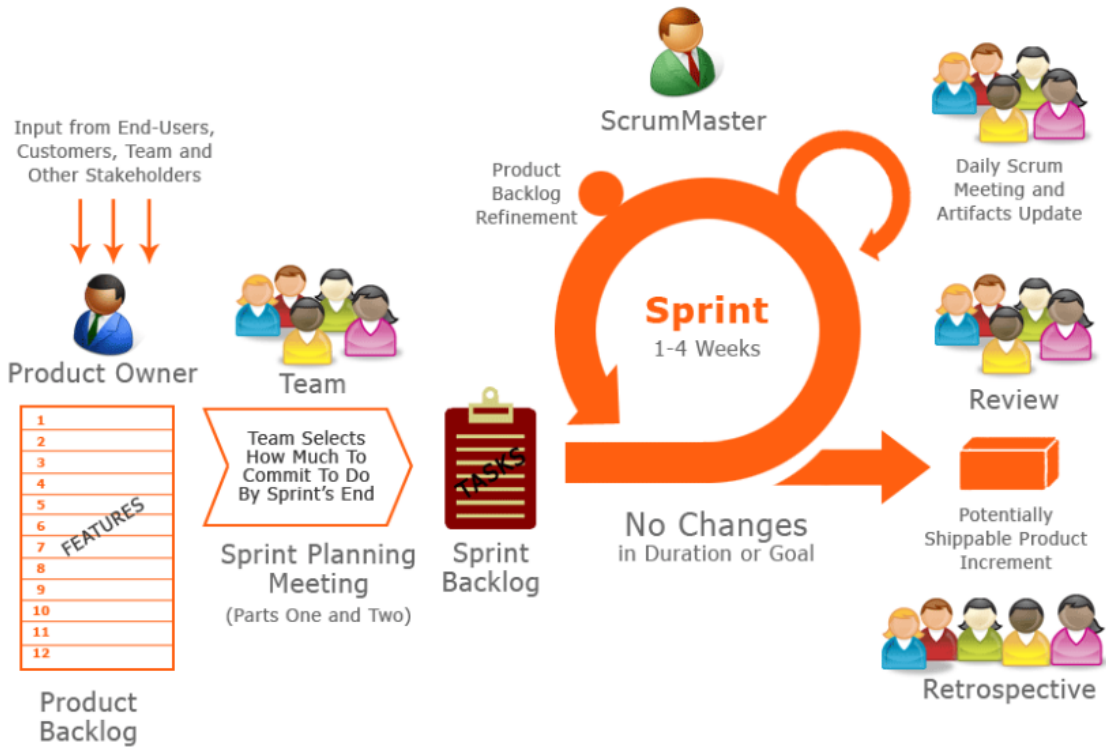
৩। আর্কিটেকচারাল গাইডলাইনের মধ্য দিয়েও এজাইল টিমে নিজেদের টেকনিকাল ডিজাইন নিরূপণ করতে সক্ষম

৪। ইটারেটিভ অথবা প্রোগ্রাম ইনক্রিমেন্ট টাইমবক্সে আবদ্ধ অবস্থায় এজাইল টিম নিজেদের কাজ সম্পূর্ণ করতে সক্ষম

৫। ডেলিভারেবল প্রডাক্টের মান উন্নতিকল্পে এজাইল টিম সর্বদাই তৎপর

৬। এজাইল টিম সর্বদাই নিজেদের উন্নতির জন্য তৎপর

নিচের চিত্রের সাহায্যে এজাইল টিম দেখানো হলঃ



৭। এজাইল এর ফ্রেমওয়ার্ক: স্ক্রাম

স্ক্রাম কি?

স্ক্রাম একটি এজাইল ম্যানেজমেন্ট ফ্রেমওয়ার্ক যা সফটওয়্যার ডেভেলপমেন্টে ব্যবহার হয়। এই পদ্ধতি ইটারেটিভ ও ইনক্রিমেন্টাল ডেভেলপমেন্ট সাপোর্ট করে। এই পদ্ধতিতে সব ধরনের সফটওয়্যার তৈরি করা যায়। স্ক্রামের একটি বিশেষত্ব হচ্ছে এতে কোন প্রজেক্ট ম্যানেজার নেই। সাধারণত প্রোডাক্ট ওনার এবং স্ক্রাম মাস্টার মিলে প্রজেক্ট ম্যানেজারের দায়িত্ব পালন করে থাকেন। স্ক্রামের সাহায্যে খুব সহজে আংশিক সফটওয়্যার ইমপ্লিমেন্ট করা যায়।

স্ক্রামের অংশগুলো হল:

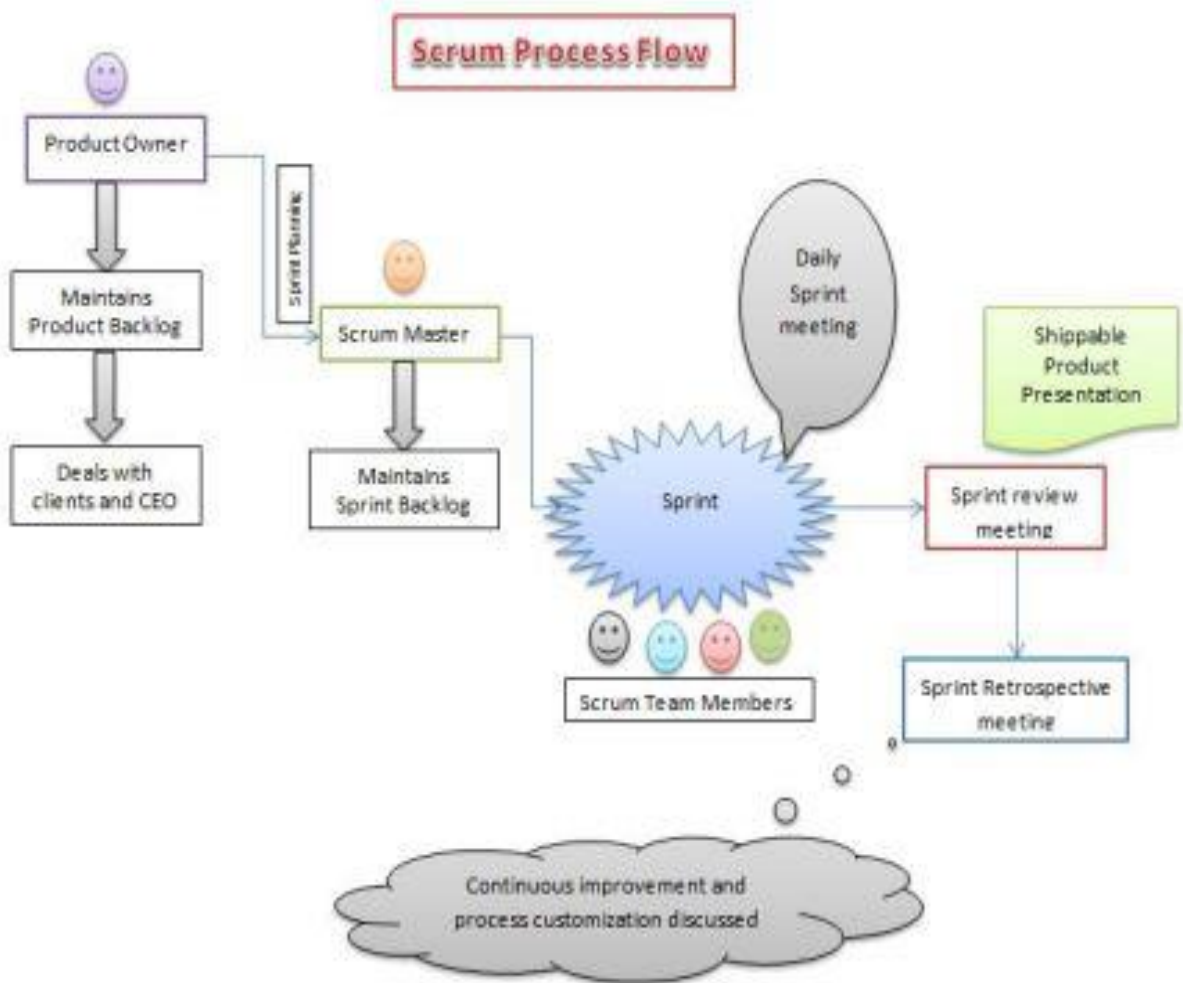
১) এর তিনটি রোল আছে: প্রোডাক্ট ওনার, স্ক্রাম মাস্টার ও স্ক্রাম টিম।

২) প্রয়োজনীয়তার ভিত্তিতে প্রোডাক্ট ব্যাকলগ।

৩) স্প্রিন্ট

৪) স্প্রিন্ট ইভেন্ট- স্প্রিন্ট প্ল্যানিং মিটিং, ডেইলি স্ক্রাম মিটিং, স্প্রিন্ট রিভিউ মিটিং, রেট্রোস্পেক্টিভ মিটিং।

স্ক্রামের ধাপ গুলো নিচের চিত্রের সাহায্যে দেখানো হল:



স্প্রিন্ট কি?

স্ক্রাম ফ্রেমওয়ার্কে স্ক্রাম প্রোডাক্ট ব্যাকলগ থেকে এন্ট্রি বাস্তবায়নের জন্য প্রয়োজনীয় সব কার্যক্রম স্প্রিন্টের মধ্যে সংগঠিত হয়। স্প্রিন্ট সবসময় সংক্ষিপ্ত হয়: সাধারণত প্রায় ২-৪ সপ্তাহ হয়।

প্রতিটি স্প্রিন্ট সেশন দুইটি স্প্রিন্ট প্ল্যানিং মিটিং এর মাধ্যমে শুরু হয়:

১। the WHAT-Meeting (কী করা হবে)

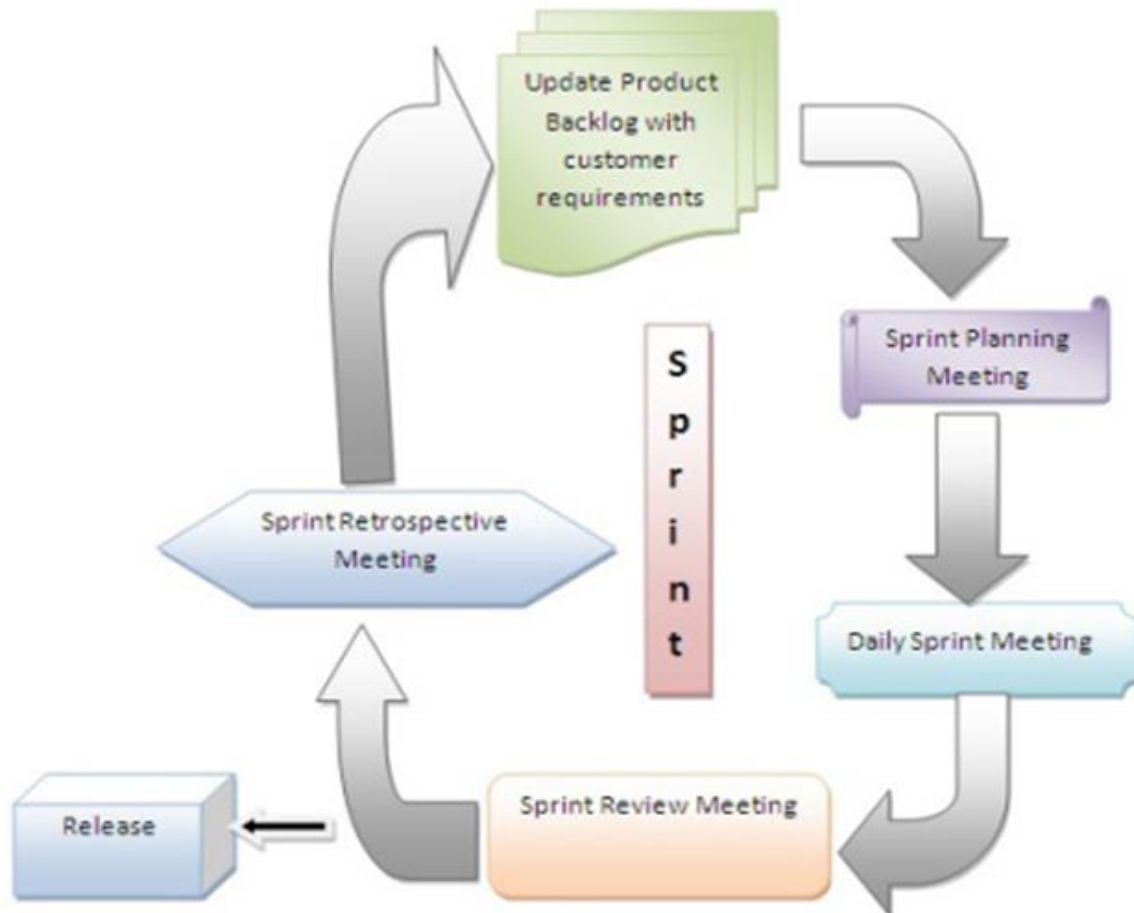
২। the HOW-Meeting (কীভাবে করা হবে)

এই দুইটি মিটিং কে সম্মিলিত ভাবে স্প্রিন্ট প্ল্যানিং মিটিং বলে। WHAT-Meeting এ বলা থাকে স্ক্রাম প্রোডাক্ট ব্যাকলগ এর কোন কোন ইউজার স্টোরি গুলো সম্পন্ন করা হবে এবং HOW-Meeting এ বলা থাকে ইউজার স্টোরিগুলো কিভাবে সম্পন্ন করা হবে।

প্রতিটি স্প্রিন্ট এর শেষে স্প্রিন্ট রিভিউ মিটিং হয় যাতে স্ক্রাম প্রোডাক্ট ওনার রিভিউ করে দেখেন যে ওই স্প্রিন্টের ইউজার স্টোরিগুলো সম্পূর্ণ রূপে এবং সঠিকভাবে শেষ হয়েছে কিনা।

সবার শেষে একটি রет্রোস্পেক্টিভ হয় যাতে স্প্রিন্টের ভাল দিক, খারাপ দিক এবং কোন কোন বিষয় গুলো আরো ভালো হওয়া দরকার তা নিয়ে আলোচনা করা হয়।

স্প্রিন্টের ধাপ গুলো নিচের চিত্রের সাহায্যে দেখানো হল:



স্ক্রাম প্রোডাক্ট ব্যাকলগ কি?

স্ক্রাম প্রোডাক্ট ব্যাকলগ হল ঐ সমস্ত কাজের লিস্ট যে সব কাজ একটি প্রজেক্টের মধ্যে সম্পন্ন করতে হয়।

এটা প্রথাগত রিকয়ারমেন্ট স্পেসিফিকেশন ডকুমেন্টকে প্রতিস্থাপন করে। প্রোডাক্ট ব্যাকলগ এর মালিক প্রোডাক্ট ওনার। প্রোডাক্ট ব্যাকলগ এ কাজ করার মানে এই নয় যে অন্য কোনো ডকুমেন্ট ব্যবহার করা যাবে না। কিন্তু অন্য যেসব ডকুমেন্ট ব্যবহৃত হয় সবগুলোই প্রোডাক্ট ব্যাকলগ এর সহায়ক ডকুমেন্ট হিসেবে কাজ করে। এই ডকুমেন্ট গুলোতে সাধারনত প্রোডাক্ট ব্যাকলগ এর বর্ণনা থাকে।

প্রোডাক্ট ওনার সাধারনত স্ক্রাম প্রোডাক্ট ব্যাকলগটি স্ক্রাম প্ল্যানিং মিটিং এর সময় ব্যবহার করে থাকেন, এর এন্ট্রি গুলোকে স্ক্রাম টিম মেম্বারদের বুঝিয়ে দেয়ার জন্য। তখন টিম মেম্বাররা সিদ্ধান্ত নেয় কোন কোন রিকয়ারমেন্টগুলো নেক্সট স্প্রিন্টে শেষ করা যাবে।

স্ক্রাম প্রোডাক্ট ব্যাকলগ এর বৈশিষ্ট্যগুলো হল:

- ১। স্ক্রাম প্রোডাক্ট ব্যাকলগ এর প্রতিটি এন্ট্রি কাস্টমারের কাছে মূল্য সংযোজন করবে
- ২। স্ক্রাম প্রোডাক্ট ব্যাকলগ এর সকল এন্ট্রি প্রয়োজনীয়তার উপর ভিত্তি করে সাজানো থাকবে
- ৩। স্ক্রাম প্রোডাক্ট ব্যাকলগ এর সকল এন্ট্রি ভালোভাবে খতিয়ে দেখা হবে
- ৪। স্ক্রাম প্রোডাক্ট ব্যাকলগ এর সকল এন্ট্রি কখনোই বিবরণ মূলক হবে না

Product Backlog Estimating System Upgrade					
Sprint	ID	Backlog Item	Owner	Estimate (days)	Remaining (days)
1	1	Minor Remove user kludge in .dpr file	BC	1	1
1	2	Minor Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	1	1
1	3	Minor Create "Legacy" discipline node with old civils and E&I content	BC	1	1
1	4	Major Augment each tbl operation to support network operation	BC	10	10
1	5	Major Extend Engineering Design estimate items to include summaries	BC	2	2
1	6	Super Supervision/Guidance	CAM	4	4
	7	Minor Remove Custodian property from AppConfig class in globals.pas	BC	1	
	8	Minor Remove LOC_ constants in globals.pas and main.pas	BC	1	
	9	Minor New E&I section doesn't have tblCaption set	BC	1	
	10	Minor Delay in main.releaseform doesn't appear to be required	BC	1	
	11	Minor Undo modifications to Other Major Equipment in formExcel.pas	BC	1	
	12	Minor AJACS form to be centred on the screen	BC	1	
	13	Major Extend DUnit tests to all 40 disciplines	BC	6	

স্ক্রাম ফ্রেমওয়ার্ক কেন সফল?

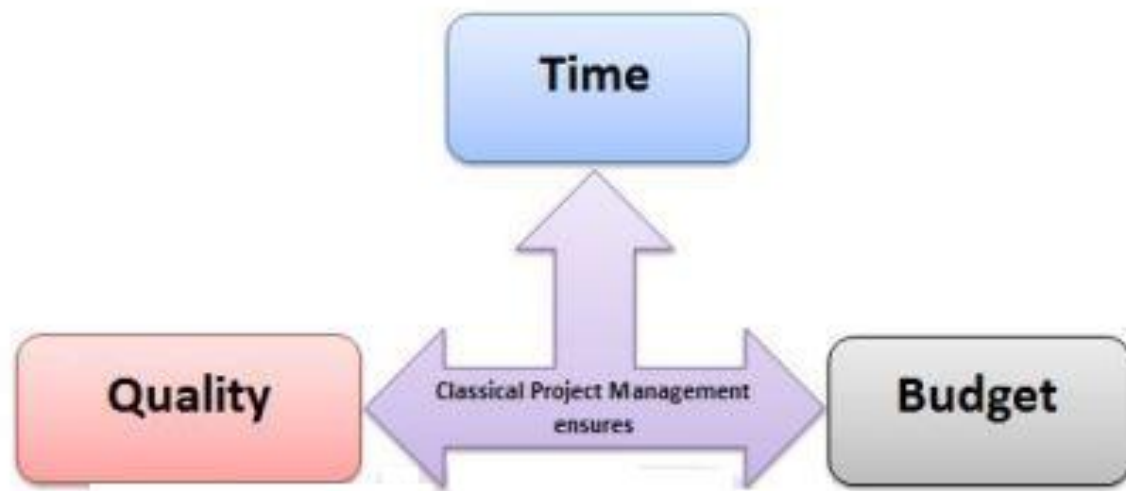
স্ক্রাম প্রকল্প ব্যবস্থাপনার ক্যালেন্ডার ত্রিভুজ পরিবর্তন করে। এটি এখন আর সময়, বাজেট এবং কোয়ালিটির মধ্যে নেই, এটা এখন বাজেট, সময় এবং কার্যকারিতার ত্রিভুজ হয়ে উঠছে।

স্ক্রাম এর মূল বৈশিষ্ট্য হল কোন অসম্পূর্ণ বা অপরিষ্কৃত বৈশিষ্ট্য ক্লায়েন্টের এর নিকট রিলিজ করা হয় না। সঠিকভাবে পরীক্ষিত, ডকুমেন্টেড ও ফিচারযুক্ত ভার্শনগুলোইগুলোই কেবল ক্লায়েন্টের নিকট যায়। এটা সাধারনত ইনক্রিমেন্টের মাধ্যমে বর্ধিত হতে থাকে।

ক্রাম মেথডের সুফলগুলো হলঃ

- ১) বর্ধিত উৎপাদন
- ২) উন্নত পণ্যের মান
- ৩) স্বল্প এবং স্থিতিশীল প্রকল্পের খরচ
- ৪) উচ্চ গ্রাহক সন্তুষ্টি
- ৫) কর্মচারীদের প্রেরণা ও সন্তুষ্টি বৃদ্ধি

ক্রাম ও ক্লাসিকাল সফটওয়ার ডেভেলপমেন্টের মধ্যে পার্থক্য চিত্রের সাহায্যে দেখানো হলঃ



৮। রেফারেন্স

- ১। https://en.wikipedia.org/wiki/Agile_software_development
- ২। <http://agilemanifesto.org/iso/en/principles.html>
- ৩। <https://www.agilealliance.org/agile101/the-agile-manifesto/>
- ৪। <http://subzerosolutions.co.uk/baan/blog/implementation-methodology/>
- ৫। <https://firstlinesoftware.com/who-we-are/agile-development>
- ৬। <http://www.ambyssoft.com/essays/agileRoles.html>
- ৭। <http://www.scaledagileframework.com/agile-teams/>