বাংলাদেশ সুইডেন পলিটেকনিক ইন্সটিটিউট

বিষয় শিক্ষকের নামঃ বিউটি বিশ্বাস

পদবীঃ ইন্সট্রাক্টর (কম্পিউটার)

প্রিন্সিপালস অব সফটওয়্যার ইঞ্জিনিয়ারিং

বিষয় কোডঃ ৬৬৬৬১

দ্বিতীয় অধ্যায়

সফটওয়্যার ডেভেলপমেনন্ট লাইফ সাইকেল এর মৌলিক বিষয়

সফটওয়্যার ইঞ্জিনিয়ারিং সম্পর্কে ধারণা

পাঠ সমূহঃ

01 সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেল কার্যক্রম

02 সফটওয়্যার ডেভেলপমেন্ট মডেলসমূহ

03 অ্যাজাইল ডেভেলপমেন্ট মডেল

04 অ্যাজাইল ডেভেলপমেন্ট প্রক্রিয়া

05 অ্যাজাইল ম্যানিফেস্টো

06 অ্যাজাইল ম্যানিফেস্টো-এর আইটেমসমূহ

07 অ্যাজাইল প্রক্রিয়া

সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেল কর্যক্রম:

❖ সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেল বা সংক্ষেপে SDLC হলো সফটওয়্যার প্রকৌশলে কাঙ্ক্ষিত পণ্য উৎপাদনের সুপরিচিত ও সুগঠিত কার্যক্রমের ধাপসমূহ। SDLC কাঙ্ক্ষিত পণ্য বা সফটওয়্যার ডিজাইন ও ডেভেলপের জন্য কয়েকটি ধাপে কাজ করে, যা নিম্নে বর্ণনা করা হলো-

- **যোগাযোগ (Communication) :** এটি প্রথম ধাপ, যেখানে গ্রাহক কোনো সফটওয়্যার-এরজন্য চাহিদা প্রকাশ করে, সফটওয়্যার সেবা প্রদানকারীর সাথে যোগাযোগ করে ও শর্তাবলি জানায়। সেবা প্রদানকারীর কাছে লিখিতভাবে তার চাহিদার কথা জানাতে হয়।
- **চাহিদা সংবদ্ধ করা (Requirement Gathering) :** সফটওয়্যার ডেভেলপার টিমের জন্য এটি প্রথম কাজ। ডেভেলপার টিম-২০২১]গ্রাহকের সাথে যত বেশি সম্ভব সমস্যা নিয়ে কথা বলে তাদের চাহিদাটা বুঝতে চায় এবং মূলত গ্রাহক চাহিদা, সিস্টেমের চাহিদা ও ফাংশনাল চাহিদা এসব ধরনের চাহিদা যত ভালোভাবে বুঝতে পারবে ততই ভালো। চাহিদাগুলো কয়েকটি নিয়ম মেনে সংবদ্ধ করা হয় –

i. অপ্রচলিত বা বিদ্যমান সিস্টেম ও সফটওয়্যার নিরীক্ষণ করে
i. ডেভেলপার ও গ্রাহকের সাক্ষাৎকার নিয়ে
ii. ডাটাবেসের সাহায্য নিয়ে
iii. উদ্ভূত প্রশ্নের সমাধান করে।

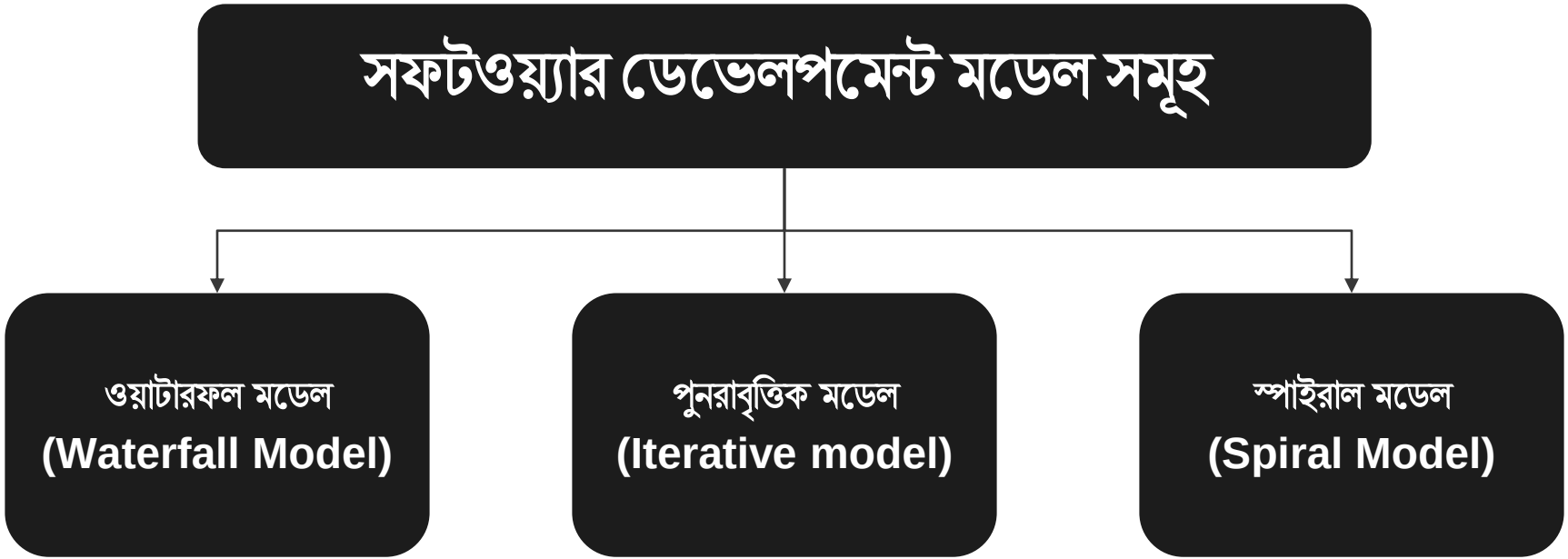
সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেল কর্যক্রম:

- **সম্ভাব্যতা নিরীক্ষণ করা (Feasiblitystudy):** চাহিদা সংবদ্ধ করার পর ডেভেলপার টিম সফটওয়্যার প্রসেসের একটি খসড়া তৈরি করে। এরপর তারা নির্ণয় করার চেষ্টা করে, সফটওয়্যারটি গ্রাহকের চাহিদা পুরোপুরি পূরণ করবে নাকি এটি ব্যবহারের অনুপযোগী, সফটওয়্যারটি অর্থনৈতিকভাবে, প্রযুক্তিগতভাবে ও ব্যবহারিকভাবে উপযুক্ত কি না। সফটওয়্যারটির সম্মাহাতা নিরীক্ষণ করার জন্য বিভিন্ন ধরনের অ্যালগরিদমিক পদ্ধতি ব্যবহার করা হয়।

- **সিস্টেম অ্যানালাইসিস (System Analysis):** এ পর্যায়ে ডেভেলপার টিম তাদের রোডম্যাপ পরিকল্পনা ঠিক করে এব প্রকল্পের জন্য সবচেয়ে উপযুক্ত মডেল নির্বাচন করে।
- **সফটওয়্যার ডিজাইন (Software Design) :** এরপর সমস্ত তথ্য ও শর্তাবলি একসাথে করে সফটওয়্যারটি ডিজাইন করতে হয়। গ্রাহক চাহিদা, শর্তাবলি ও প্রয়োজনীয় তথ্য, যা চাহিদা সংবদ্ধ করার সময় পাওয়া গিয়েছিল সেগুলোই এই ধাপে ইনপুট আউটপুট দুটি ধাপে আসবে লজিক্যাল ডিজাইন ও ফিজিক্যাল ডিজাইন। প্রকৌশলীরা কিছু মেটা ডাটা, ডাটা ডিকশনারি, লজিক্যা ডায়াগ্রাম, ডাটা ফ্লো ডায়াগ্রাম এবং কিছু কেসে কিছু সিউডোকোড প্রস্তুত করেন।

সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেল কর্যক্রম:

- **কোডিং করা (Coding) :** এ ধাপও প্রোগ্রামিং ধাপ হিসেবে ধরা হয়। সঠিকভাবে উপযুক্ত প্রোগ্রামিং ভাষায় নির্ভুলভাবে প্রোমান কোড লেখার মাধ্যমে সফটওয়্যার ডিজাইনের নিরীক্ষণকার্যক্রম বাস্তব রূপ লাভ করে।
- **নিরীক্ষণ (Testing):** একটি সমীক্ষায় দেখা গেছে, ৫০% সফটওয়্যার ডেভেলপমেন্ট নিরীক্ষণ করা হয়। সামান্য ভুলের জন্য সফটওয়্যারের ক্ষতি এমনকি বাতিলও হতে পারে। ডেভেলপাররা কোডিং-এর সময় মডিউল টেস্টিং, প্রোগ্রাম টেস্টিং, অভ্যন্তরীণ টেস্টিং, গ্রাহকভিত্তিক টেস্টিং ইত্যাদি নিরীক্ষণ করেন।
- **মৌলিকতা (Integration) :** লাইব্রেরি ডাটাবেস ও অন্যান্য প্রোগ্রামের ক্ষেত্রে সফটওয়্যারের মৌলিকতা থাকতে হবে। SDLC- এই বৈশিষ্ট্যই বিশ্ব বাজারে এর স্বাতন্ত্র্য নিশ্চিত করবে।
- **প্রয়োগ (Implementation) :** গ্রাহকের যন্ত্রে সফটওয়্যার ইনস্টল করতে হবে। ইনস্টল করার পরে কনফিগার করতে হয়, এরপর বহনযোগ্যতা, অভিযোজন ও মৌলিকতা নির্ণয় করে দেখা হয় ।
- **কার্যক্রম ও রক্ষণাবেক্ষণ (Operations a Maintenance) :** এ পর্যায়ে অধিক দক্ষতা ও কম ভুলের মাধ্যমে কার্যক্রম চলে। গ্রাহক পর্যায়ে পরিবেশ বা প্রযুক্তিগত পরিবর্তন অনুসারে কোড আপডেট করে সফটওয়্যার রক্ষণাবেক্ষণ করা হয়, যার মাধ্যমে লুকানো ক্ষতিকর বিষয় ও বাস্তব জীবনের অপ্রত্যাশিত সমস্যা মোকাবেলা করা সম্ভব হয় ।
- **প্রবণতা (Disposition) :** সময়ের সাথে সাথে সফটওয়্যারের সক্ষমতা কমতে থাকে। এটি একদম অকেজো হয়ে পড়ে বা আপডেটের দরকার হয় এবং কোনো বড় অংশ বাছাই করার দরকার হতে পারে। ডাটা ও সফটওয়্যারের কোনো অংশ একসাথে রাখা, সিস্টেম বন্ধ করা, প্রবণতার পরিকল্পনা, কাজ শেষে সিস্টেম ত্যাগ করা ইত্যাদি এ ধাপের অন্তর্গত।


সফটওয়্যার ডেভেলপমেন্ট মডেল সমূহ
ওয়াটারফল মডেল
(Waterfall Model)
পুনরাবৃত্তিক মডেল
(Iterative model)
স্পাইরাল মডেল
(Spiral Model)

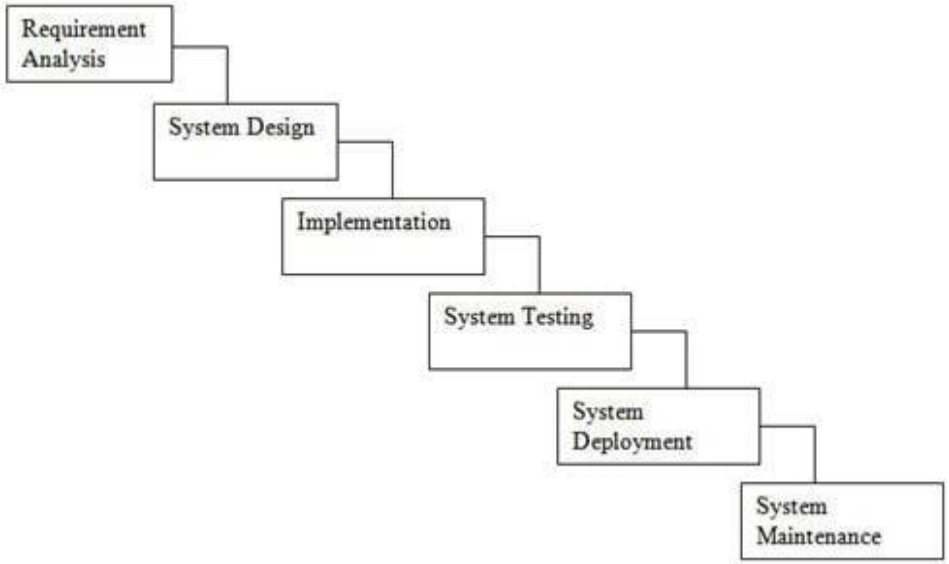
ওয়াটারফল মডেল (Waterfall model):

সফটওয়্যারআলোচনা করা হলো- ডেভেলপমেন্ট মডেলের মধ্যে সবচেয়ে সহজ মডেল হলো ওয়াটারফল মডেল। এটি রৈখিক নিয়ম মেনে চলে। এ মডেল অনুসারে SDLC প্রক্রিয়ার এক পর্ব শেষ হলে তবেই অন্য আরেক পর্ব শুরু হয়। এই মডেল সকল ধাপ সুষ্ঠুভাবে হয়েছিল কি না তা নিশ্চিত করে। এক পর্বের কাজ চলার সময় পূর্ববর্তী কোনো পর্বের কথা মাথায় রাখার দরকার পড়ে না।

(i) প্রয়োজনীয়তা অনুধাবন (Requirement Analysis): একটি সফটওয়্যার তৈরি করতে কোন কোন জিনিস প্রয়োজন তার একটা নথি তৈরি করা।

(ii) সিস্টেম ডিজাইন (System Design) : প্রথম ধাপে যে প্রয়োজনীয় বৈশিষ্ট্যাবলি একটা নথিতে অন্তর্ভুক্ত করা হয়েছিলতার উপর ভিত্তি করে একটা সফটওয়্যার ডিজাইন করতে হবে।

(iii) বাস্তবায়ন (Implementation) : সিস্টেম ডিজাইন থেকে ইনপুট নিয়ে, সিস্টেমটি প্রথমে ছোট ছোট অংশে ভাগ করতে হবে। প্রতিটি অংশের কার্য সম্পূর্ণ করতে হবে এবং পরবর্তী পর্যায়ে উক্ত অংশগুলোকে একীভূত করতে হবে।



(iv) সিস্টেম পরীক্ষা (System Testing): সিস্টেম/সফটওয়্যার তৈরি করার পর কিনা তা যাচাই সফটওয়্যারটিতে কোনো সমস্যা/ত্রুটি আছেকরা।

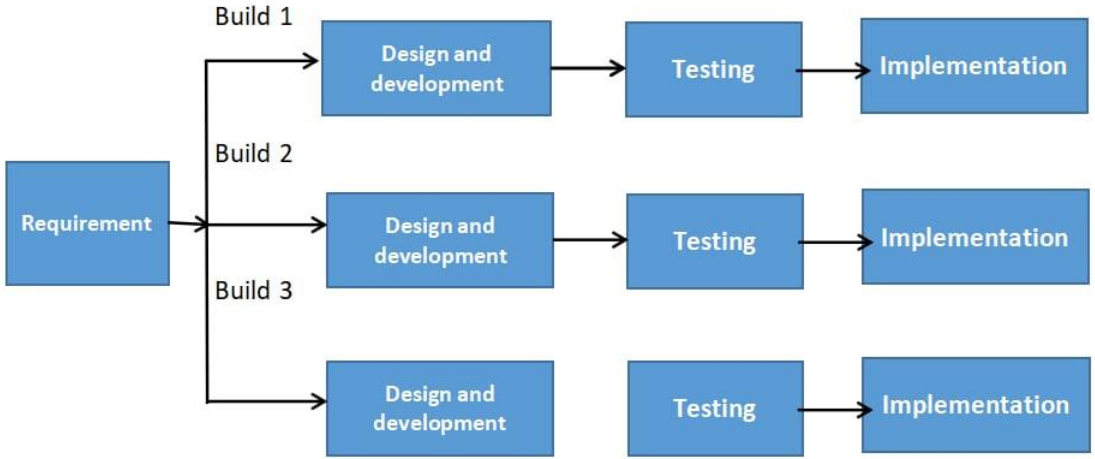
(v) সিস্টেম স্থাপন (System Deployment): সিস্টেম/সফটওয়্যারটি কার্যকর না অকার্যকর তা পরীক্ষা করা হয় এ স্তরে।সিস্টেমটি যদি গ্রাহকের নিকট কার্যকর বলে মনে হয় তাহলে সিস্টেমটি বাজারে ছাড়া হয় ।

(vi) সিস্টেম রক্ষণাবেক্ষণ (System Maintenances): প্রতিটি সিস্টেমে কিছু সমস্যা থাকে, যা ক্লায়েন্ট-এর কাছ থেকে পরবর্তীতে সংগ্রহ করা হয়। উক্ত সমস্যাগুলোকে নিয়ে সিস্টেমটিকে আপডেট করতে হবে। সে কারণে পুরোনো সিস্টেম/সফটওয়্যারকে রক্ষণাবেক্ষণ করতে হবে।

তবে এই মডেলে সকল ধাপ নিখুঁতভাবে সম্পন্নকরতে হয়। পূর্বের কোনো কাজ অসমাপ্ত রাখা এই মডেলে সম্ভব নয়। ডেভেলপার একই ধরনের সফটওয়্যারে পূর্বে কাজ করেছে এ রকম ক্ষেত্রে ওয়াটারফল মডেল অনুসরণ করা যায়।

পুনরাবৃত্তিক মডেল (Iterative model):

এই মডেলে ক্ষুদ্র পরিসর তৈরি করা হয় এবং যৌক্তিক সকল পন্থা অনুসরণ করাহয়। এরপর পরবর্তী সকল চক্রে নতুন অনেক ফিচার ও মডিউলকে ডিজাইন, কোড ও টেস্ট করা হয়। তারপর সফটওয়্যারে যোগ করা হয়। প্রতি চক্রের পরে ব্যবস্থাপনা টিম পরবর্তী চক্রের জন্য তৈরি হতে পারে। যেহেতু প্রতিটি চক্র ছোট ছোট অনেক ভাগে বিভক্ত, তাই ব্যবস্থাপনা প্রক্রিয়া অপেক্ষাকৃত সহজ। তবে এতে বেশি রিসোর্সের প্রয়োজন হয়।



স্পাইরাল মডেল (Spiral model) :

সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেলের অন্যতম গুরুত্বপূর্ণ মডেল হলো স্পাইরাল মডেল। স্পাইরাল মডেল ডায়াম্যাটিক উপস্থাপনা, যা দেখতে অনেকটা লুপের মতো। সফটওয়্যার ডেভেলপমেন্টে প্রক্রিয়ার প্রতিটি লুপকে এক একটি ফেজ বলা হয়। স্পাইরাল মডেলের ব্যাসার্ধের উপর নির্ভর করে প্রোজেক্টের খরচ কত হবে তা জানা যায়। ব্যাসার্ধ বেশি হলে। খরচ বেশি হয় এবং ব্যাসার্ধ কম হলে খরচ কম হয়। সফটওয়্যার-এর অগ্রগতি নির্ভর করে কৌণিক মাত্রার উপর।

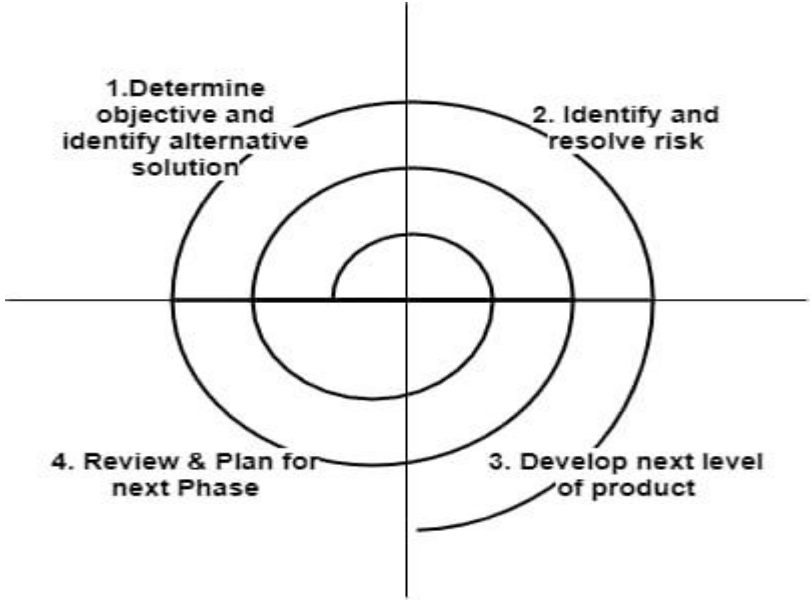
(i)উদ্দেশ্য নির্ধারণ: গ্রাহকদের কাছ থেকে প্রয়োজনীয় তথ্য সংগ্রহ করাই এই ভাগের মূল উদ্দেশ্য।

(ii)ঝুঁকি চিহ্নিতকরণ: দ্বিতীয় চতুর্ভাগে সময় সম্ভাব্য সর্বোত্তম সমাধান নির্বাচন করার জন্য সমস্ত সম্ভাব্য সমাধান মূল্যায়ন করা হয়। তারপর সেই সমাধানের সাথে যুক্ত ঝুঁকিগুলো চিহ্নিত করা হয়।

সিস্টেমের পরবর্তী সংস্করণ তৈরিকরণ: সফটওয়্যারটিতে যে

(iii)বৈশিষ্ট্যগুলো থাকার কথা তা আছে কি না সেটি যাচাইকরতে হবে। যদি না থাকে বা নতুন কোনো বৈশিষ্ট্য সংযোজন করতে হয় তাহলে সফটওয়্যারটির পরবর্তী সংস্করণ করতে হবে।

(iv)পরবর্তী সংস্করণ এবং পরিকল্পনা তৈরি: গ্রাহকরা সফটওয়্যারটি মূল্যায়ন করবে এবং সেই সাপেক্ষে বিভিন্ন তথ্য দিবে, যা পরবর্তী সংস্করণের জন্য কাজে লাগবে।।



অ্যাজাইল ডেভেলপমেন্ট মডেল

পুনরাবৃত্তিক (Iterative) ওয়াটারফল মডেলের প্রধান সমস্যা হলো প্রকল্প উন্নয়নের মধ্যকালে গ্রাহকদের কাছ থেকে বিভিন্ন প্রকার পরিবর্তনের অনুরোধ এবং এই পরিবর্তনগুলোকে অন্তর্ভুক্ত করার জন্য প্রয়োজনীয় উচ্চ মূল্য এবং সময়। ওয়াটারফল মডেলের এই ত্রুটি দূর করতে ১৯৯০ দশকের মাঝামাঝি সময়ে অ্যাজাইল সফটওয়্যার ডেভলপমেন্ট মডেল প্রস্তাব করা হয়। অ্যাজাইল মডেলের মূল উদ্দেশ্য কোনো প্রকল্পকে দ্রুত এবং সহজে সমাপ্ত করা।

অ্যাজাইল ডেভেলপমেন্ট প্রক্রিয়া

অ্যাজাইল সফটওয়্যার ডেভেলপমেন্ট মডেলে প্রথমে একটি কাজকে ছোট ছোট অংশে ভাগ করা হয়। অ্যাজাইল মডেল পুনরাবৃত্তিমূলক (Iterative) ডেভেলপমেন্ট প্রক্রিয়ায় কাজ করে। প্রতিটি অংশকে একেকটি পুনরাবৃত্তিমূলক (Iteriative) প্রক্রিয়ায় ভাগ করা হয় এবং ক্রমবর্ধমান (Incremental) অংশসমূহ সমাধান করা হয়। একসাথে শুধুমাত্র একটি ইটারেশন পরিকল্পনা করা হয় এবং সম্পন্ন করে গ্রাহকদের কাছে উপস্থাপন করা হয়। অ্যাজাইল মডেল পুনরাবৃত্তিমূলক এবং ক্রমবর্ধমান প্রক্রিয়ার সমন্বিত মডেল।অ্যাজাইল মডেলের ধাপগুলো হলো—

(i) প্রয়োজনীয়তা সংগ্রহ (Requirement gathering)
(ii) প্রয়োজন বিশ্লেষণ এবং পরিকল্পনা (Requirement analysis and plan)
(iii) ডিজাইন (Design)
(iv) কোডিং (Coding)
(v) অংশ যাচাইকরণ (Unit testing)
(vi) স্বীকৃতি যাচাইকরণ (Acceptance testing)।

অ্যাজাইল ম্যানিফেস্টো

অ্যাজাইল ম্যানিফেস্টো হলো একপ্রকার ঘোষণা বা ডিক্লারেশন, যা ৪টি মূল্যমান (Values)-এর সাথে ১২টি নীতির (Principles) বর্ণনা দেয়। ২০০১ সালের ফেব্রুয়ারির ১১-১৩ তারিখে ১৭ জন ডেভেলপারের সমন্বয়ে অ্যাজাইল ম্যানিফেস্টো তৈরি করা হয়। অ্যাজাইল ম্যানিফেস্টো ডেভেলপারগণকে অ্যাজাইল অ্যালাইন্স (Agil alliance) বলা হয়।

অ্যাজাইলম্যানিফেস্টোর অবস্থা (State of the agile manifesto) : অ্যাজাইল সফটওয়্যার ডেভেলপমেন্টের চারটি অবস্থা রয়েছে-

(i) ব্যক্তি, প্রসেস এবং tools-এর মধ্যে সমন্বয়সাধন।

(ii) ব্যাপক ডকুমেন্টের উপর ভিত্তি করে সফটওয়্যারের কার্যপরিধি নির্ভর করে।

(iii) চুক্তি মোতাবেক গ্রাহকদের সহযোগিতা প্রদান ।

(iv) একটি পরিকল্পনা অনুসরণ করে পরবর্তী সফটওয়্যার পরিকল্পনা।

অ্যাজাইল ম্যানিফেস্টো-এর আইটেম সমূহঃ

২০০১ সালের ফেব্রুয়ারিতে ইউটাহ-এর স্নোবার্ড রিসোর্টে ১৭ জন ডেভেলপার একত্রিত হন লাইটওয়েট ডেভেলপমেন্ট নিয়ে আলোচনা করার জন্য। তাঁদের সম্মেলনের ফলে সফটওয়্যার ডেভেলপমেন্টের অ্যাজাইল ম্যানিফেস্টোর ঘোষণা আসে, যা ৪টি মূল্যমান (Values)-এর সমন্বয়ে গঠিত।

অ্যাজাইল ম্যানিফেস্টো ডেভেলপারগণ বা অ্যালাইন্সদের ভাষ্যমতে, "আমরা এটি করে এবং অন্যকে এটি করতে সাহায্য করে সফটওয়্যার ডেভেলপমেন্টের জন্য ভালো উপায় বের করছি। এ কাজের মাধ্যমে আমরা যে বিষয়গুলো মূল্যায়ন করব সেগুলো হলো-

(i) ব্যক্তি ও যোগাযোগকে প্রক্রিয়া ও যন্ত্রপাতির উপরে মূল্যায়ন করব।
(ii) ওয়ার্কিং সফটওয়্যারকে বিশদ ডকুমেন্টেশনে উপরে মূল্যায়ন করব।
(iii) গ্রাহক সম্পর্ককে আলাপ-আলোচনা বা চুক্তির উপরে মূল্যায়ন করব।
(iv) পরিবর্তনের সাথে তাল মেলানোকে প্ল্যান অনুসরণের উপরে মূল্যায়ন করব।

অর্থাৎ বাম পাশের আইটেমগুলোকে ডান পাশের আইটেমের তুলনায় বেশি মূল্য দিতে হবে।" এই চারটি মূল্যমানকে (Values) অ্যাজাইল ম্যানিফেস্টো-এর আইটেমও বলা হয়।

Individual and interactions	over	Process and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

অ্যাজাইল ম্যানিফেস্টো-এর মূলনীতি:

অ্যাজাইল ম্যানিফেস্টো-এর ৪টি মূল্যমান (Values) এর সাথে ১২টি মূলনীতি (Principles) রয়েছে। নিম্নে ১২টি মূলনীতি প্রদান করা হলো:

(i) গ্রাহক সন্তুষ্টি
(ii) পরিবর্তনকে স্বাগত জানানো
(iii) ওয়ার্কিং সফটওয়্যার উৎপাদন করা
(iv) সমন্বয়সাধন
(v) মোটিভেশন
(vi) মুখোমুখি কথোপেকথন
(vii) কাজের অগ্রগতি পরিমাপ করা
(viii) কনস্ট্যান্ট প্রা মেইন্টেন করা
(ix) মনিটরিং করা
(x) সরল টার্ম ব্যবহার
(xi) স্বনিয়ন্ত্রিত টিম
(xii) নিয়মিত কাজ রিভিউ করা

অ্যাজাইল প্রক্রিয়া (Agile Methods):

১.ডাইনামিক সিস্টেম ডেভেলপমেন্ট মেথড (Dynamic system development method):ডিএসডিএম (DSM) সফটওয়্যার প্রকল্পের জন্য উপযোগী অ্যাজাইল কাঠামো। গতানুগতিক পদ্ধতির সাথে মিলাতে এটি তৈরা করা হয়েছে। ডিএসডিএম-এর সর্বশেষ সংস্করণকে বলে ডিএসডিএম অ্যাটার্ন।

২.অ্যাজাইল ড্রাম মেথড (Agile scrum method) : দলগত ডেভেলপমেন্ট পরিবেশে এটি বেশি উপযোগী। ক্রাম ইন্টার‍্যাক্সিড ইনক্রিমেন্ট মডেল ব্যবহার করে এবং ইটারেশনের সংক্ষিপ্ত রূপ ব্যবহার করে।

৩. এক্সট্রিম প্রোগ্রামিং (Extreme programming): এক্সট্রিম' প্রোগ্রামিং সফটওয়্যার ডেভেলপমেন্টের একটি ধারা, যেখানে কর্মীদের অধিক মানসম্পন্ন সফটওয়্যার বেশি উৎপাদনশীলতার সাথে উৎপাদনে উৎসাহিত করে। এক্সপি যে বিষয়গুলো নিয়ে কাজ করে সেগুলো পরীক্ষামূলকহলো বিশ্লেষণ, ডেভেলপমেন্ট এবং নতুনভাবে উদ্ভাবিত ধারণার ক্ষেত্র, যা কিনা চূড়ান্তভাবে উৎপাদিত পণ্যের অন্তর্নিহিত মানের তুলনা করে।

অ্যাজাইল প্রক্রিয়া (Agile Methods):

৪. টেস্ট-ড্রিভেন ডেভেলপমেন্ট (Test-driven development): টিডিডি (TDD) বা টেস্ট-ড্রিভেন ডেভেলপমেন্ট একপ্রকারসফটওয়্যার ডেভেলপমেন্ট প্রক্রিয়া, যেটিতে অত্যন্ত ক্ষুদ্র ডেভেলপমেন্টের চক্র ঘন ঘন পুনরাবৃত্তি করা হয়।

৫. লিন এবং কানবান সফটওয়্যার ডেভেলপমেন্ট (Lean and Kanban software development) : এটি একটি উৎপাদনপ্রক্রিয়া, যেখানে কোনো লক্ষ্যের জন্য রিসোর্স বরাদ্দ করা হয় চূড়ান্ত পর্যায়ে গ্রাহকের অপচয় রোধের জন্য। লিন অল্প কাজে ভ্যালুর সঞ্চয়- এই মূলনীতির উপর নির্ভর করে।

এই অধ্যায় থেকে যা যা শিখেছি:

- সফটওয়্যার ডেভেলপমেন্ট লাইফ সাইকেল কর্যক্রম।
- সফটওয়্যার ডেভেলপমেন্ট মডেল সমূহ।
- ওয়াটারফল মডেল (Waterfall Model)
- পুনরাবৃত্তিক মডেল (Iterative model)
- স্পাইরাল মডেল (Spiral Model)
- অ্যাজাইল ডেভেলপমেন্ট মডেল

ধন্যবাদ