

	<u>Waterfall</u>	<u>RAD</u>	<u>Evolutionary</u>	<u>Prototype</u>	<u>Spiral</u>	<u>Scrum</u>
1. Мащаб на проекта (големи, средни, малки)			От средни до големи.	От средни до големи.	При големи проекти.	От малки до големи проекти.
2. Приблизително времетраене на жизнения цикъл	Дълъг жизнен цикъл, поради линейността на модела.	Основна цел на модела е кратък жизнен цикъл. 60-90 дни	Няма ограничения.	Няма ограничения.	Поне няколко месеца.	Поне 1 месец
3. Консумация на човешки ресурси (много хора/ малко хора)	Поради линейността на модела в конкретен момент се ползват сравнително малко служители.	Консумация на много човешки ресурси, поради приложения pipelining.	Слаба консумация в началото до средна консумация след това, поради възможността за наличие на pipelining.	Може да консумират много ресурси.		Малко на брой хора.
4. Характер на човешките ресурс (само senior или може и стажанти от ФМИ)	Поради лесната разбираемост на модела, той е приложим във разнороден екип от специалисти. Както опитни, така и неопитни.	Разчита на по-опитни програмисти, запознати с всички реализирани в миналото софтуерни компоненти.	Екипът трябва да има много добри комуникационни умения.	При еволюционния прототип е желателно присъствието на опитни архитекти и проектанتي.	Анализатори на риска са нужни.	Предимно опитни експерти с умения във всички фази на софтуерния жизнен цикъл. Добри комуникативни умения.
5. Гъвкавост/ реактивност спрямо промени (променяем/ непроменяем)	Липсва механизъм за реакция на промени в изискванията. Подходящ при предварително ясни и стабилни изисквания..	Поради компонентното разделяне на софтуер, моделът е сравнително гъвкав по отношение на промени в изискванията.	В инкременталния вариант на този модел софтуерът е лесно променяем за дребни и средни (неархитектурни) промени, поради сравнително модулираната структура на софтуера.	Изменяем поради модулността в архитектурата на софтуера.		Гъвкав спрямо изменения. Това е част от идеологията на модела.
6. Ангажираност на ЗЛ по време на проекта.	ЗЛ са слабо ангажирани - само при началното изготвяне на изискванията.	Слаба до средна ангажираност.	Силна ангажираност.	Много силна ангажираност.		Много силна ангажираност. Това е основният проблем на този модел.
7. Гъвкавост на изискванията (Стабилни/ Ясни/ Променливи)						
8. Критичност на системата (жизнена важност/ бизнес важност/ маловажна)			Първоначално създадените модули са силно изтествани.			
9. Предвидимост и проследяване	Силна предвидимост.	Единственият риск за предвидимостта	Много трудно проследяване.	Трудно проследяване.		Средна до силна предвидимост,

		са проблеми в интеграцията на отделните компоненти.	Опасност от score creep.			благодарение на backlog-а и периодичната оценка на прогреса.
10. Разбираемост на модела	Лесно разбираем.	Средна разбираемост. Проблематично може да е детайлното разбиране на интеграцията.	Средна разбираемост. Проблематично може да е детайлното разбиране на интеграцията.		Труден за разбиране.	Труден за разбиране.
11. Документация	Пълна формална документация.					Средно тежко документиране.
12. Финансова оценка на проекта	Тъй като в различни етапи от проекта са ангажирани различни служители, то има периоди, в които за някои от тях няма работа. От това дали на тези служители им се плаща за времето, в което не работят, зависи фин. оценка на проекта.	Скъп при ползването на нови технологии, поради липсата на реализирани в предни проекти компоненти. Скъп и поради високата консумация на човешки ресурс. При преизползване на стари компоненти цената пада чувствително.	Ако се стигне до score creep, може да стане скъп. Но от друга страна поради честата комуникация с клиента рискът от некоректно или излишно създадени функционалности е по-нисък, което сваля от финансовата оценка.	Може да стане скъп.		Поради наличието на добри специалисти в Scrum екипа, финансите нужни за проекта може да изглеждат много. Но това се изплаща със сравнително високо качество на продукта и удовлетвореност от клиента, за когото няма изненади накрая.
13. Въздействие върху архитектурата на софтуера.	Няма.	Архитектурата трябва да е компоненто-базирана.	В инкременталния модел СА трябва да е силно декомпозирана.	При еволюционния прототип честите промени по прототипа и първоначалната липса на яснота по отношение на изискванията може да доведе до лоша архитектура.	Архитектура, ориентирана към риска. При проектирането ѝ първо се дефинират и декомпонират рисковете. Архитектурата се съобразява с тях.	Силна модулираност на архитектурата, за да може тя да се раздели на части в backlog-а и съответните спринтове.
14. Приложение	Ясни и непроменяеми изисквания	Ресурсите не са от значение. Важна е бързината на изработка.	При липса на човешки ресурс в началните етапи. В зависимост от обратната връзка се определя дали да се продължи с проекта. Приложим при високо рискови проекти.	Когато изискванията не са ясни. Налични са добри проектанتي, които да приложат множество тактики за архитектурна изменяемост на еволюционния прототип. Финансовите ресурси не са силно ограничени.	Изискванията не са ясни. Много рискове.	Когато има възможност за интензивна комуникация с клиента и се разполага със силен и балансиран екип от софтуерни инженери.