Softwareentwicklung

Was zeichnete die Software-Krise aus?
 : r1 Projektverlängerung und Preiserhöhung, geringe Qualität, Schwierigkeit bei Wartung und Innovation, schlechte Arbeitsproduktivität
 : r2 Mangel an Programmierern
 : r3 steil billigere Computersysteme
 : r4 keiner der aufgelisteten
: r1 ok 2
--
Was ist nicht die Ursache für die Software-Krise?
 : r1 Erweiterung und Erhöhung der Projektkosten
 : r2 nicht verwaltete Technologien
 : r3 Unterschätzung von Bedrohungen und Risiken
 : r4 Unkenntnis der Grundregeln
: r1 ok 2
--

"Eine Disziplin, die sich mit echten Softwareentwicklungsproblemen befasst", ist eine Definition
 : r1 Disposition Phase

 :r2 Development Phase

 :r3 Integration and Test Phase

 :r4 Operations and Maintenance phase

: r1 ok 2
--
Was ist die letzte Phase des SDLC-Software-Lebenszyklus?
 : r1 Entsorgungsphase
 : r2 Entwicklungsphase
 : r3 Integrations- und Testphase
 : r4 Betriebs- und Wartungsphase
: r1 ok 2
--

Was sind nicht die Nachteile eines Wasserfallansatzes?
 : r1 Wiederholen
 : r2 Unmöglichkeit, die resultierende Produktqualität während der Entwicklung abzuschätzen
 : r3 Abhängigkeit des Endprodukts von der Eingangsqualität
 : r4 Entwicklungszeit zu lang
: r1 ok 2

--

Der Ansatz, bei dem jede Entwicklungsaktivität periodisch wiederholt wird und bei jeder Wiederholung ein kleiner Satz von Funktionen hinzugefügt wird, die zum Zielzustand führen, wird aufgerufen
 : r1 iterativer Ansatz
 : r2 Wasserfall Ansatz
 : R3 agiler Ansatz
 : r4 "Exploratory" Programmieransatz
: r1 ok 2
--

Menschen und ihre Interaktionen sind wichtiger als Prozesse und Werkzeuge. Arbeitssoftware ist wichtiger als detaillierte Dokumentation, die Zusammenarbeit mit dem Kunden ist wichtiger als die abgeschlossenen Verträge; auf Veränderungen zu reagieren ist wichtiger als sich an den Plan zu halten - es sind Prinzipien
 : r1 agiler Ansatz
 : r2 iterativer Ansatz
 : r3 Wasserfall Ansatz
 : r4 explorative Programmieransatz
: r1 ok 2
--
Die Ermittlung der wichtigsten Systemfunktionen, die wichtigsten Anwendungsfälle, ist eines der Ziele:
 : r1 Startphase
 : r2 Entwicklungsphase
 : r3 Bauphase
 : r4 Bereitstellungsphase
: r1 ok 2
--

Die Ausgabe der Ausarbeitungsphase ist:
 : r1 ausführbare, getestete Architektur (Arbeitsteil der Anwendung)
 : r2 Verständnis der Probleme und identifizierten Risiken.
 : r3 Beta-Release-Anwendung
 : r4 Produkt bereit für die endgültige Bereitstellung
: r1 ok 2
--
Die Ausgabe der Bauphase ist
 : r1 Beta-Release-Anwendung
 : r2 ausführbare, getestete Architektur (Arbeitsteil der Anwendung)
 : r3 Produkt bereit für die endgültige Bereitstellung
 : r4 Verständnis der Probleme und identifizierten Risiken.
: r1 ok 2

--

Die Ausgabe der Übergangsphase ist:
  : r1 Produkt bereit für die endgültige Bereitstellung
  : r2 Verständnis des Problems und identifizierte Risiken.
  : r3 ausführbare, getestete Architektur (Arbeitsteil der Anwendung)
  : r4 Beta-Release-Anwendung
: r1 ok 2
--
Nach prof. Vondrák höchste zeitliche Zuordnung?
  : r1 Schöpfung
  : r2 Entwicklung
  : r3 Start
  : r4 Übertragung
: r1 ok 2
--

Der Konsens der Interessengruppen über Umfang, Kosten und Zeitplan des Projekts sowie die Einigung über die Schätzung aller Risiken und Risikominderungsstrategien sind zwei der grundlegenden Kriterien:
  : r1 LOM
  : r2 RUP
  : r3 UML
  : r4 SDLC
: r1 ok 2
--
Event, Activity und Gateway sind Elemente
  : r1 fließen Objekte
  : r2 Objekte verbinden
  : r3 Schwimmbahnen
  : r4 Artefakte
: r1 ok 2

--

Ein Diagramm mit Instanzen von Klassen und den Beziehungen zwischen ihnen wird aufgerufen
 : r1 Objektdiagramm
 : r2 Klassendiagramm
 : r3 Komponentendiagramm
 : r4 strukturdiagramm
: r1 ok 2
--
Ein Diagramm zur Modellierung von rechnerischen, organisatorischen Prozessen oder Datenflüssen heißt:
 : r1 Aktivitätsdiagramm
 : r2 Verhaltensdiagramm
 : r3 Komponentendiagramm
 : r4 Objektdiagramm
: r1 ok 2
--

Welches Diagramm wird am häufigsten zur Veranschaulichung von Kunden-System-Beziehungen verwendet?
 : r1 Anwendungsfalldiagramm
 : r2 Aktivitätsdiagramm
 : r3 Verhaltensdiagramm
 : r4 Komponentendiagramm
: r1 ok 2
--
Folgendes wird verwendet, um Klarheit und Konsistenz zu prüfen:
 : r1 testet die Dokumentation
 : r2 Sicherheitstests
 : r3 Stresstests
 : r4 Usability-Tests
: r1 ok 2
--

Definieren, Messen, Analysieren, Verbessern und Verwalten sind Phasen
 : r1 DMAIC
 : r2 LOM
 : r3 LAC
 : r4 UML
: r1 ok 2
--
Mit welchem ​​Test überprüfen wir die Funktionsweise der einzelnen Programmzweige?
 : r1 White-Box-Test
 : r2 Black-Box-Test
 : r3 Verifikation
 : r4 Auswertung
: r1 ok 2