

Software Development WBS Example

This is an example of a work breakdown structure (WBS) for a traditional waterfall approach to software development.

- 1 Requirements Definition (Phase 1)
 - 1.01 Requirements funding
 - 1.01.01 Review project request
 - 1.01.02 Establish preliminary justification
 - 1.01.03 Fund Phase 1
 - 1.01.04 Prioritize project
 - 1.01.05 Establish project team
 - 1.02 Define problem or opportunity
 - 1.02.01 Interview users
 - 1.02.02 Examine operation and reports
 - 1.02.03 Document problem from symptoms
 - 1.02.04 Define project scope
 - 1.03 Analyze existing system
 - 1.03.01 Assemble documentation on existing system
 - 1.03.02 Identify data flows
 - 1.03.03 Identify external interfaces
 - 1.03.04 Identify problem domain
 - 1.04 Document system requirements
 - 1.04.01 Document output information requirements
 - 1.04.02 Document interface constraints
 - 1.04.03 Document audit trail constraints
 - 1.04.04 Document turnaround or response time
 - 1.04.05 Document security constraints
 - 1.04.06 Document physical environment constraints or requirements
 - 1.05 Validate against systems architecture
 - 1.05.01 Compare requirements with systems architecture
 - 1.05.02 Identify possible inconsistencies
 - 1.05.03 Identify conflicting and concurrent development
 - 1.05.04 Identify recommended changes to systems architecture
 - 1.05.05 Identify recommended changes to strategic implementation plan
 - 1.06 Management review / Phase 2 funding
 - 1.06.01 Plan next phase
 - 1.06.02 Reevaluate development costs
 - 1.06.03 Reevaluate justification
 - 1.06.04 Obtain user requirements consensus
 - 1.06.05 Obtain technical consensus
- 2 Logical Design (Phase 2)
 - 2.01 Identify detailed data requirements
 - 2.01.01 Identify output requirements
 - 2.01.02 Decompose output data
 - 2.01.03 Identify input requirements
 - 2.01.04 Identify sources of input
 - 2.02 Develop prototype or user system view
 - 2.02.01 Design interactive screens
 - 2.02.02 Design reports
 - 2.03 Design database
 - 2.03.01 Define logical data relations
 - 2.03.02 Design data structure
 - 2.03.03 Validate database design
 - 2.04 Structure processes
 - 2.04.01 Isolate highly related data
 - 2.04.02 Reconstruct processes to correspond with output requirements

Software Development WBS Example

- 2.04.03 Identify reusable process structures
- 2.05 Design Interfaces
 - 2.05.01 Design external data interfaces
 - 2.05.02 Design human interfaces
 - 2.05.03 Design intersystem interfaces
- 2.06 Specify all inputs and outputs
 - 2.06.01 Define data and interface relations
 - 2.06.02 Define data and system relations
- 2.07 Develop preliminary test and conversion procedures
 - 2.07.01 Identify test requirements
 - 2.07.02 Create test checklist
 - 2.07.03 Identify conversion requirements
 - 2.07.04 Create conversion checklist
- 2.08 Validate logical design
 - 2.08.01 Validate data relations
 - 2.08.02 Validate process relations
 - 2.08.03 Validate process logic
 - 2.08.04 Trace requirements to design
- 2.09 Validate against system architecture
 - 2.09.01 Compare logical design with systems architecture
 - 2.09.02 Identify possible inconsistencies
 - 2.09.03 Identify conflicting and concurrent development
 - 2.09.04 Identify recommended changes to systems architecture
 - 2.09.05 Identify recommended changes to strategic implementation plan
- 2.10 Management review / Phase 3 funding
 - 2.10.01 Plan next phase
 - 2.10.02 Reevaluate development costs
 - 2.10.03 Reevaluate justification
 - 2.10.04 Obtain user requirements consensus
 - 2.10.05 Obtain technical consensus
- 3 Physical Design (Phase 3)
 - 3.01 Design or specify physical database
 - 3.01.01 Review logical database design
 - 3.01.02 Determine access methods to be used
 - 3.01.03 Normalize database
 - 3.01.04 Design database architecture
 - 3.01.05 Identify reusable database structures
 - 3.01.06 Develop detailed database layout
 - 3.01.07 Develop database file, record, and schema descriptions
 - 3.01.08 Develop module calling sequences
 - 3.01.09 Update data dictionary entries
 - 3.01.10 Validate physical database design
 - 3.02 Design processing structure
 - 3.02.01 Compose process structures from data decomposition
 - 3.02.02 Identify physical subsystems
 - 3.02.03 Identify physical programs
 - 3.02.04 Identify reusable programs
 - 3.02.05 Eliminate process redundancies
 - 3.02.06 Develop teleprocessing network specifications
 - 3.03 Design processing logic
 - 3.03.01 Design calling sequences
 - 3.03.02 Develop calculation specifications
 - 3.03.03 Design interface logic
 - 3.03.04 Design security logic
 - 3.03.05 Design error recovery logic

Software Development WBS Example

- 3.04 Define procedures
 - 3.04.01 Review logical system design user interfaces
 - 3.04.02 Develop interactive data-entry procedures
 - 3.04.03 Develop screen specifications
 - 3.04.04 Design input forms
 - 3.04.05 Develop physical report specifications
 - 3.04.06 Develop user operating procedures
 - 3.04.07 Develop data processing operations run procedures
- 3.05 Refine test/conversion procedures
 - 3.05.01 Review test/conversion plans
 - 3.05.02 Update test/conversion plans
- 3.06 Validate physical design
 - 3.06.01 Validate data relations
 - 3.06.02 Validate process relations
 - 3.06.03 Validate process logic
 - 3.06.04 Validate procedures
 - 3.06.05 Validate teleprocessing network specifications
 - 3.06.06 Validate system timing and sizing requirements
- 3.07 Validate against systems architecture
 - 3.07.01 Compare physical design with systems architecture
 - 3.07.02 Identify possible inconsistencies
 - 3.07.03 Identify conflicting / concurrent development
 - 3.07.04 Identify recommended changes to systems architecture
- 3.08 Management review / Phase 3 funding
 - 3.08.01 Plan next phase
 - 3.08.02 Reevaluate development costs
 - 3.08.03 Reevaluate justification
 - 3.08.04 Obtain user requirements consensus
 - 3.08.05 Obtain technical consensus
- 4 Programming and Unit Testing (Phase 4)
 - 4.01 Decompose program modules
 - 4.01.01 Identify program modules
 - 4.01.02 Identify program module input and output
 - 4.01.03 Identify reusable modules
 - 4.01.04 Eliminate module redundancies
 - 4.02 Develop program modules
 - 4.02.01 Develop detailed module logic
 - 4.02.02 Validate module logic
 - 4.02.03 Code module
 - 4.02.04 Develop module test data
 - 4.02.05 Develop call and called stubs
 - 4.02.06 Unit test program
 - 4.03 Update test/conversion procedures
 - 4.03.01 Review test/conversion plans
 - 4.03.02 Update test/conversion plans
 - 4.04 Management review / Phase 3 funding
 - 4.04.01 Plan next phase
 - 4.04.02 Reevaluate development costs
 - 4.04.03 Reevaluate justification
 - 4.04.04 Obtain user requirements consensus
 - 4.04.05 Obtain technical consensus
- 5 System Testing (Phase 5)
 - 5.01 Finalize integrated system test plan
 - 5.01.01 Review interim test procedures
 - 5.01.02 Develop integration test procedures

Software Development WBS Example

- 5.01.03 Develop integration test plan
 - 5.01.04 Assign integration test responsibilities
 - 5.01.05 Develop integration test data
 - 5.01.06 Train data processing personnel
- 5.02 Finalize user acceptance/training test plan
 - 5.02.01 Review interim test procedures
 - 5.02.02 Develop user acceptance criteria
 - 5.02.03 Develop final user acceptance test procedures
 - 5.02.04 Develop user acceptance test plan
 - 5.02.05 Assign user acceptance test responsibilities
 - 5.02.06 Develop user acceptance test data
 - 5.02.07 Train users
- 5.03 Conduct integration test
 - 5.03.01 Link programs and copy to test libraries
 - 5.03.02 Establish test files
 - 5.03.03 Execute integration test
- 5.04 Conduct user acceptance/training test
 - 5.04.01 Establish user acceptance test files
 - 5.04.02 Establish test files
 - 5.04.03 Execute user acceptance test
- 5.05 Management review / Phase 3 funding
 - 5.05.01 Plan next phase
 - 5.05.02 Reevaluate development costs
 - 5.05.03 Reevaluate justification
 - 5.05.04 Obtain user requirements consensus
 - 5.05.05 Obtain technical consensus
- 6 Installation (Phase 6)
 - 6.01 Finalize conversion plan
 - 6.01.01 Review interim conversion procedures
 - 6.01.02 Develop conversion procedures
 - 6.01.03 Develop conversion plan
 - 6.01.04 Assign conversion responsibilities
 - 6.02 Convert files/database
 - 6.03 Install software
 - 6.04 Run systems in parallel
 - 6.05 Turn off old system
 - 6.06 Management review
 - 6.06.01 Obtain user requirements consensus
 - 6.06.02 Obtain technical consensus
 - 6.06.03 Document recommended changes to architecture
 - 6.06.04 Document recommended enhancements to system