EVA Analysis

8 tasks have been completed but the project schedule indicates that 11 tasks should have been completedin that time.

Task	Planned effort		Actual effort.	
1	21.0		22.5	
2	13.0		17.0	
3	19.0		23.8	
4	15		18.2	
5	20		22.0	
6	14.0		16.0	
7	15.0		17.0	
8	7.0		6.5	
9	12.8			
10	4.6			
11	8.6			

BCWP

BCWS

ACWP

So here, BAC = 346

BCWS = 150

BCWP = 124

ACWP = 143

SPI = BCWP / BCWS = 124 / 150= 0.82

SV = BCWP – BCWS = 124 – 150 = -26 person-day

CPI = BCWP / ACWP = 124 / 143= 0.87

CV = BCWP – ACWP = 124 – 143= -19 person-day

% Schedule for completion = BCWS / BAC

= 150/346

= 43.90%

[% of work schedule to be done at this time]

% Complete = BCWP / BAC

= 124 / 346

= 35.83%

[% of work completed at this time]

Risk Analysis

Risks	Category	Probability	Impact
Size estimated might be lower than expectation	PS	55%	2
Number of users might be higher than expectation	PS	25%	3
Larger number of users than planned	PS	35%	3
Less reuse than planned	PS	70%	2
Deviation from define software development process	PR	40%	2
Delivery might exceed deadline	BU	45%	2
Funding will be lost	CU	40%	1
Project budget might exceed expectation	CU	40%	1
Data Security Breach	TE	25%	2
Personnel shortfalls	ST	20%	4
Developing the wrong software functions	TE	5%	1
Developing the wrong user interface	TE	5%	1
Late changes to requirements	BU	30%	3
Development technically too difficult	ST	10%	2

Security Vulnerabilities	TE	30%	2
Inexperienced Staff	ST	35%	2
Hardware Failure	TE	15%	2

Risks	Risk reduction technique
Size estimated might be lower than expectation	Conduct thorough requirements gathering and analysis to ensure accurate estimation. Use techniques such as story points or function points for more precise size estimation. Involve domain experts and stakeholders in the estimation process to validate assumptions.
Number of users might be higher than expectation	Perform scalability testing to ensure the system . Implement load balancing techniques to distribute user traffic effectively. Monitor system performance regularly and upgrade hardware or infrastructure as needed.
Less reuse than planned	Develop reusable components or libraries to maximize code reuse. Establish coding standards and guidelines to encourage modular and reusable code. Conduct code reviews to identify opportunities for reuse and refactor code when necessary.
Deviation from define software development process	Provide comprehensive training and support to ensure adherence to the defined process. Implement process automation tools to streamline development workflows and enforce process compliance. Conduct regular process audits and reviews to identify and address deviations proactively.
Delivery might exceed deadline	Break down project tasks into smaller, manageable chunks and set realistic deadlines for each. Use project management techniques such as Agile or SCRUM to facilitate iterative development and prioritize tasks effectively. Continuously monitor progress and adjust plans as needed to mitigate schedule risks.
Funding will be lost	Diversify funding sources to reduce dependency on a single funding stream. Establish contingency funds or reserves to mitigate the impact of funding loss. Maintain open communication with stakeholders to address funding concerns proactively.

Project budget might exceed expectation	Conduct thorough cost estimation and analysis to identify potential budget risks upfront. Implement cost-tracking mechanisms to monitor expenses and identify budget overruns early.

Data Security Breach	Implement encryption techniques to protect sensitive data both at rest and in transit. Implement access controls and authentication mechanisms to restrict unauthorized access to system resources.
Personnel shortfalls	Cross-train, recruit proactively, and offer training.
Developing the wrong user interface	Conduct usability testing and involve UX experts.
Late changes to requirements	Implement change management processes to evaluate and prioritize requirement changes effectively. Conduct impact assessments to understand the implications of requirement changes on project scope, schedule, and budget.
Development technically too difficult	Conduct feasibility studies and technical assessments upfront to identify potential challenges. Break down complex tasks into smaller, more manageable components.
Security Vulnerabilities	Implement secure coding practices and conduct regular code reviews to identify and fix security vulnerabilities. Utilize penetration testing and vulnerability scanning tools to assess the system's security posture.
Inexperienced Staff	Provide comprehensive training and mentoring programs to onboard inexperienced staff. Pair less experienced team members with seasoned mentors to facilitate knowledge transfer.
Hardware Failure	Use reliable hardware components from reputable manufacturers with a proven track record. Implement redundancy and failover mechanisms to mitigate the impact of hardware failures. Conduct regular maintenance and performance testing to identify and address potential hardware issues proactively.