

Software Release Planning as an Intelligent Agent Problem

Backlog → Planning Agent → Release Plan

Goal of the project:

- Model release planning as an **optimization problem**
- Use a **Genetic Algorithm (GA)** – based agent to propose a high-value release plan
- Compare against a **simple greedy baseline**

Synthetic Software Backlog & Optimization Objectives

- Synthetic backlog with 30 features, inspired by real Product items
- Categories: Security & Compliance, Integrations & Data, Usability & UX etc.
- Each feature has:
 - effort (story points)
 - business_value (1–10)
 - must_have flag
 - strategic_flag
- Release capacity: TEAM_CAPACITY = 50 story points
- Chromosome: 30-bit vector, 1 = include feature, 0 = exclude.
- Fitness function:
$$\text{fitness} = \text{value_score} - W_{\text{must}} \cdot \text{must_penalty} - W_{\text{cap}} \cdot \text{cap_penalty}$$
 - value_score: normalized total value
 - must_penalty: fraction of missing must-haves
 - cap_penalty: normalized over-capacity
 - Weights:
 - W_MUST_HAVE = 2.0
 - W_CAPACITY = 1.0

Synthetic Software Backlog

id	title	category	effort	business_value	must_have	strategic_flag	dependencies
F01	Encrypt configuration backups and disable plaintext exports	Security & Compliance	5	9	1	1	[]
F02	Improve session dashboard load time for large deployments	Reliability & Performance	3	7	0	0	[]
F03	Introduce zero-trust access policy engine for field IEDs	Security & Compliance	5	9	1	1	[]
F04	Add quick-connect bookmarks for frequently accessed substations	Usability & UX	2	8	0	0	[]
F05	Normalize Crossbow asset IDs with PowerBase equipment model	Integrations & Data	5	9	1	1	[]
F06	Predefined access policies for AEP operational roles	Customer-Specific	3	8	0	1	["F03"]
F07	Predefined access policies for FirstEnergy operations team	Customer-Specific	3	8	0	1	["F03"]
F08	Export full historical session logs to long-term archive	Internal / Tech Debt	8	4	0	0	[]
F09	Centralized security audit log pipeline to utility SIEM	Security & Compliance	3	8	1	1	[]
F10	Highlight critical alarms on remote-access launch screen	Usability & UX	2	9	0	1	[]
F11	Visual diff for device configuration snapshots	Usability & UX	5	7	0	0	["F01"]
F12	Unified firmware inventory collector for multi-vendor IEDs	Integrations & Data	3	8	1	1	[]
F13	Sync firmware metadata from Crossbow to PowerBase	Integrations & Data	8	7	0	1	["F05"]
F14	Sync cybersecurity posture KPIs to PowerBase dashboards	Integrations & Data	5	8	0	1	["F05"]
F15	Add dark theme option for Crossbow web UI	Usability & UX	2	6	0	0	[]
F16	Auto-expiring time-bounded access tokens for field maintenance	Security & Compliance	5	7	0	0	["F03"]
F17	Support IPv6-only deployment scenarios for remote sites	Reliability & Performance	8	3	0	0	[]
F18	Single Sign-On integration with corporate identity provider	Security & Compliance	5	8	1	1	[]
F19	Add context help links to device detail pages	Usability & UX	3	6	0	0	[]
F20	Pre-check connectivity to jump hosts before starting a session	Reliability & Performance	2	7	0	0	[]
F21	Firmware end-of-support risk report for all managed assets	Security & Compliance	8	7	0	1	["F12"]
F22	Role-mapping wizard for SSO groups to Crossbow roles	Security & Compliance	3	7	0	1	["F18"]
F23	Clean up stale offline devices from inventory in the background	Internal / Tech Debt	5	6	0	0	[]
F24	Enforce MFA for all internet-facing Crossbow portals	Security & Compliance	3	8	1	1	[]
F25	Auto-generate firmware upgrade recommendation bundles	Reliability & Performance	8	7	0	1	["F12"]
F26	Add in-app link to Crossbow user documentation portal	Usability & UX	2	5	0	0	[]
F27	Tag devices by substation criticality (Tier 1/2/3)	Security & Compliance	5	6	0	0	[]
F28	Bulk-edit metadata for devices in the same substation	Usability & UX	3	5	0	0	[]
F29	Audit trail for SSO login failures and lockouts	Security & Compliance	8	7	0	1	["F18"]
F30	Expose Crossbow health status via REST API for NOC dashboards	Integrations & Data	5	6	0	0	[]

Two Agent Planners: Greedy Heuristic vs Genetic Algorithm

Genetic Algorithm

- Representation: 30-bit chromosome
- Initialization: random population (80 individuals)
- Selection: tournament selection ($k = 3$)
- Variation:
 - One-point crossover (rate 0.8)
 - Bit-flip mutation (rate 0.02)
- Elitism: best individual carried over each generation
- Run for 150 generations, tracking best fitness

Greedy baseline

- Step 1: Include all must-have features
- Step 2: For remaining features:
 - $\text{Score} = (\text{business_value} / \text{effort}) + \text{small bonus for strategic items}$
 - Sort by score and add while staying within capacity
- Represents a simple “PM heuristic”: fill the release with best value/effort items after must-haves

Results: Release Plans from Greedy vs GA

Metric	Greedy baseline	Genetic Algorithm
Selected features	15 / 30	16 / 30
Total effort	48	51
Over capacity	0	1
Total value	117	124
Missing must-haves	0	0
Fitness	0.5571	0.5705

GA Convergence and Future Extensions

- GA quickly improves from negative fitness (random plans) to ≈ 0.57
- Converges by generation 15–20 and then stabilizes

Limitations & Next Steps

Current model:

- Optimizes value, must-haves, and capacity

Possible extensions (future work):

- Add dependency violation penalties (child feature selected without its parent).
- Incorporate multi-release planning (Release 1 vs Release 2).
- Explore alternative selection/crossover operators GA

